Peer Assessments (https://class.coursera.org/androidapps101-001/human_grading/)

/ Assignment 3 Submission

://class.coursera.org/androidapps101-001/help/peergrading?url=https%3A%2F

oursera.org%2Fandroidapps101-001%2Fhuman_grading%2Fview%2Fcourses%2F402%2Fassessments%2F9%2Fresults%2Fmine)

Submission Phase

1. Do assignment ☑ (/androidapps101-001/human_grading/view/courses/402/assessments/9/submissions)

Evaluation Phase

2. Evaluate peers ☑ (/androidapps101-001/human_grading/view/courses/402/assessments/9/peerGradingSets)
3. Self-evaluate ☑ (/androidapps101-001/human_grading/view/courses/402/assessments/9/selfGradingSets)

Results Phase

4. See results ☑ (/androidapps101-001/human_grading/view/courses/402/assessments/9/results/mine)

Your effective grade is 10

Your unadjusted grade is 10, which was calculated based on a combination of the grade you received from your peers and the grade you gave yourself.

See below for details.

Be sure you have read the instructions on the **Assignment 3 (https://class.coursera.org/androidapps101-001/wiki/Assignment3)** page thoroughly and have completed all the materials in the checklists below before submitting your apps.

Check that you've completed all of the above steps and then gather the following materials to have ready to upload at the link below:

- The signed apk file.
- A screenshot of your app.
- Briefly answer the following 3 questions:
  - What does your app do?
  - Why did you decide to build the app?
  - What do you remember most about your development experience? For example, what was the hardest part of this assignment or the part that required the most time?
- Your complete Java file of your Activity. For submission, you can post the contents of your Java file directly into the text box.

When reviewing the work submitted by a fellow peer, install and play with the apps they provide. Give the app developer some constructive advice and some helpful feedback on their efforts.

.

Upload the signed apk file here. Note apk files *can* be uploaded even if not listed below.

happybirthdayvalentine (https://s3.amazonaws.com/coursera-uploads/user-519c50709447eca98fc650d0/402/asst-9/30e1dbf091a011e3b6c3dd077736c589.apk)

Upload a screenshot of your app here.

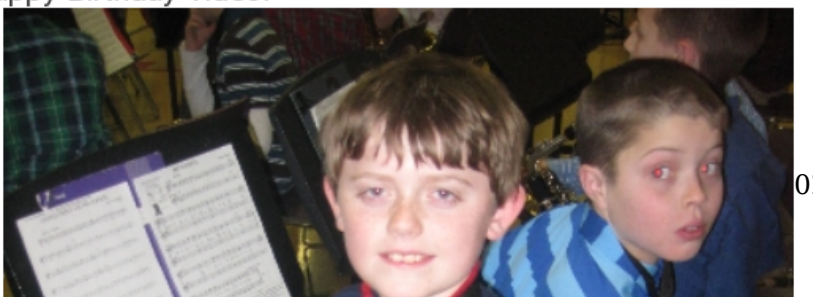
Happy Birthday Valentine!

Click in image to animate it! Happy 11th Birthday on Feb. 14, 2014, Peter! You were a Valentine's Day gift to Mom, Dad and your sister!

incognitodiscforums@gmail.com

Click! Send Email!

Click Image! Happy Birthday Video!



Briefly answer the following 3 questions.

1. What does your app do?

This app uses the new concepts learned in weeks 6 and 7 combined with some concepts we learned in earlier weeks. It uses: ScrollView, LinearLayout, OnClickListener, TextView, ImageView, Simple Animation, Intents, EditText and a Button for email. If you click on the first picture, of Peter and Sophie with party hats and noise makers in their mouths, it animates, moves to the right and then comes back. If you click on the Send Email button, you can send a happy birthday email to my incognitodiscforums@gmail.com (mailto:incognitodiscforums@gmail.com) email account. If you hit the ESC key, you come back to the app. You hear the Beatles "Happy Birthday Song - Live Version" playing in the background once the app opens, and only stops to send an email, or if you click the bottom image, of Peter with his saxophone. If you click that image, it takes you to a You Tube video of a monkey singing happy birthday. If you hit the ESC key, you come back to the app, and the Beatles happy birthday song starts to play again.

2. Why did you decide to build the app?

I decided to build this app based on the suggestions made, one of which was an app for Valentine's Day that uses the concepts learned in weeks 6 and 7 and before. Given it is my eldest child, Peter's, 11th birthday this Valentine's Day 2014, that is the reason for the theme of my app.

3. What do you remember most about your development experience?

What I remember most about my development experience, was the trouble I had setting up the Intel 10 480-800mdpi emulator to use an email client. It just would not work for security reasons or something I did not have time to figure out, and when you click the send email button on that emulator, it gives you the toast message to "please set up your email client". On the Nexus 7-18-tvdpi emulator, the email client set up procedure worked just fine, so if you click the send email button on that emulator, it works fine. However, another thing I noticed, when you click the 2nd image of Peter on the saxophone, it takes you to a You Tube video of a monkey singing happy birthday. On the Intel 10 you can view the video of the monkey singing as well as hear it, but I noticed on the Nexus 7, you could only hear the monkey but you could not see the video of the monkey singing, and I did not have time to figure out why. It works fine otherwise though, and if you hit the ESC key, you go back to the app, and you hear the Beatles raw mp3 singing happy birthday again, which plays non-stop while you are in the app, unless you either send an email (on Pause called) or if you go to You Tube (on Pause called) but once you go back to the app, it

resumes (onResume). The rest of the app's design was straight forward based on the professor's video presentations for week 6 and 7 and earlier weeks as well.

Post the contents of the complete Java file of your Activity directly into the text box below.

```
package com.example.happybirthdayvalentine;
//Karen West, Assignment #3 Android Apps Class, Feb. 9th, 2013
//This app demonstrates the use of concepts learned in Weeks 6 and 7
//and some concepts from earlier weeks as well.
//I used: ScrollView, LinearLayout, OnClickListener, TextView, ImageView
//Visibility, Simple Animation, Intents, EditText and a button for email
//(from the list the professor gave,
//where we must use at least 4 items on that list).
//I tested it on the emulator: Intel 10 480x800-mdpi and it worked fine.
//I tested it on the emulator: Nexus-7-18tvdpi, and it worked fine there too,
//except that when you went to the youtube video (clicking on bottom photo),
//the music played, but you could not see the monkey singing it, as you could
//on the other emulator, and I did not have time to find out why.
//NOTE: for the email part - the Intel 10 email client set up DID NOT work -
//I could NOT set up the email on the emulator.  However it did work on the
//Nexus 7.  Here is the set-up instructions I followed, since I do not yet own
//a mobile device and rely on the emulators for this class.
//http://www.androidaspect.com/2012/06/how-to-send-email-from-android-emulator.html
(http://www.androidaspect.com/2012/06/how-to-send-email-from-android-emulator.html)
//These worked on the Nexus 7, so that is where that part will work, with email.
//On the Intel 10 emulated device, if you click send email, you will get the
//toast message to please configure your email device, and when I tried to do that
//there as I did on the Nexus 7, it said it was not allowed on this for some reason.
//However, sending email did work on the Nexus 7.
//The Nexus 7 did NOT work though as mentioned for VIEWING the monkey singing
//happy birthday you tube video - you could hear him but not see him.
//In order to SEE the monkey on you tube singing happy birthday, you have to use the
//Intel 10 device - did not have time to figure out why.


import android.app.Activity;
import android.content.Intent;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.animation.Animation;
```

```
import android.view.animation.AnimationUtils;
import android.widget.EditText;
import android.widget.Toast;


public class MainActivity extends Activity {

    private static final String TAG = "HappyBirthdayValentineActivity";
    private EditText mEmail;

    MediaPlayer beatlesHappyBirthdayMediaPlayer;
    String url_happybirthdayvalentine = "http://www.youtube.com/watch?v=rur5wKoua1o&
list=PLA9525CE7B38D5742&index=1" (http://www.youtube.com/watch?v=rur5wKoua1o&
list=PLA9525CE7B38D5742&index=1);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mEmail = (EditText) findViewById(R.id.email);
        OnClickListener listener = new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent i = new Intent(Intent.ACTION_VIEW);

                //video not allowed on mobiles
                //change to a photo or happy birthday message
                i.setData(Uri.parse(url_happybirthdayvalentine));
                startActivity(i);
            }
        };
        findViewById(R.id.peterwithsaxophone).setOnClickListener(listener);

    }

    public void sendValentineBirthdayEmail(View v) {
        Log.e(TAG, "sendEmail");

        String email = mEmail.getText().toString();
        String message = null;

        if (email.length() > 0) {
            message = message + "\nAlternative Email:" + email;
        }

//From Professor Lawrence Angrave:
        // FYI There's lots of discussion about email intents on StackOverflow
        // eg SEND vs SENDTO and setting the mimetype to message/rfc822
```

```
        // Experimentally the following works on many devices
        // Tested on Android 1.6 and 4.x phones, and tablets and 2.x,4.x
        // emulator.
        // You will need to configure the emulator's email client with a
        // real email address.

        // To test unsupported schemes change "mailto" to "horseback"
        Log.e(TAG, "creating email Intent");
        Intent emailIntent = new Intent(Intent.ACTION_SENDTO);
        emailIntent.setData(Uri.fromParts("mailto",
              "incognitodiscforums@gmail.com (mailto:incognitodiscforums@gmail.com)", null));
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "happy birthday valentine!");

        emailIntent.putExtra(Intent.EXTRA_TEXT, message);

    //From Professor Lawrence Angrave:
        // Better .... use resolveActivity
        // We can check to see if there is a configured email client
        // BEFORE trying to start an activity
        // Using this test we could have prevented the user from ever opening
        // the survey...
        Log.e(TAG,"email intent to resolve activity");
        if (emailIntent.resolveActivity(getPackageManager()) == null) {
           Log.e(TAG, "toast to please configure email client");
           Toast.makeText(getApplicationContext(),
                 "Please configure your email client!", Toast.LENGTH_LONG).show();
        } else {
           // Secondly, use a chooser will gracefully handle 0,1,2+ matching
           // activities
           Log.e(TAG, "start activity create chooser");
           startActivity(Intent.createChooser(emailIntent,"Please choose your email app!"));
        }
    }

    @Override
    protected void onResume() {
       Log.e(TAG, "onResume!");
       beatlesHappyBirthdayMediaPlayer = MediaPlayer.create(this, R.raw.the_beatles_birthday_song);
       beatlesHappyBirthdayMediaPlayer.setLooping(true);
       beatlesHappyBirthdayMediaPlayer.start();
       super.onResume();
    }
    @Override
    protected void onPause() {
       Log.e(TAG, "onPause!");
       beatlesHappyBirthdayMediaPlayer.stop();
       beatlesHappyBirthdayMediaPlayer.release();
       super.onPause();
    }
```

```
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
    public void animatePeterAndSophie(View v) {
        Log.e(TAG, "animatePeterAndSophie");

        View view = findViewById(R.id.peterandsophie_newyears);
        // Declare the variable (the pointer!) The next line does NOT create a
        // new animation object - all we have is pointer so far.
        Animation anim;

        // Create a new animation object
        anim = AnimationUtils.makeOutAnimation(this, true);

        // Tell the view it's time to start animating
        view.startAnimation(anim);
    }
}
```

MainActivity (https://s3.amazonaws.com/coursera-uploads/user-519c50709447eca98fc650d0/402/asst-9/0f82b2c091a211e3b6c3dd077736c589.java)

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Grading Questions

- Does the app require any permissions other than INTERNET and/or WRITE_EXTERNAL_STORAGE to install? If yes: Do not continue. The review stops immediately and the project is awarded 0 points.
- Does the app implement at least 4 topics introduced in Week 6 or Week 7? (see the beginning of this assignment for a list of topics).
- Did the user upload a screenshot?
- Was an apk and the activity Java file included?
- Did the participant answer the 3 questions?
- For open feedback: Is the code commented? Can you find comments that explain how or why the code works? Do you have any advice on how to improve the app?

Assign a score to the app based on the following scale:

- **10 out of 10 points—Excellent:** The assignment is fully completed: The student

demonstrated a competency and used many (at least 4) topics introduced in Week 6 and/or Week 7. No more than 1 error or omission was found.

- **8 out of 10 points—Sufficient:** The assignment is almost complete. At least 2 errors or significant omissions were found, but these were simple oversights and could be fixed fairly quickly.
- **5 out of 10 points—Partial:** The assignment was partially completed but had significant errors and would still require significant time to finish.
- **0 out of 10 points—Poor:** The assignment fell far short of being complete or sufficient.

Score from your peers: **10**

Score from yourself: **10**

Optionally, Include any feedback you have about the app and the posted code.

self → Although I graded myself with 10 here because it met all requirements, it did not go beyond what the professor taught in lectures. I know some people in the class who have either more prior experience or they have more time to explore on their own sometimes go beyond what the professor does in lecture videos, and that is fun to peer review and learn from too. However, my app did meet all requirements so I gave it a 10.

peer 1 → Nice job of combining techniques from the class, and Happy Birthday to Peter!

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*