

Android System Tools

Tool Types

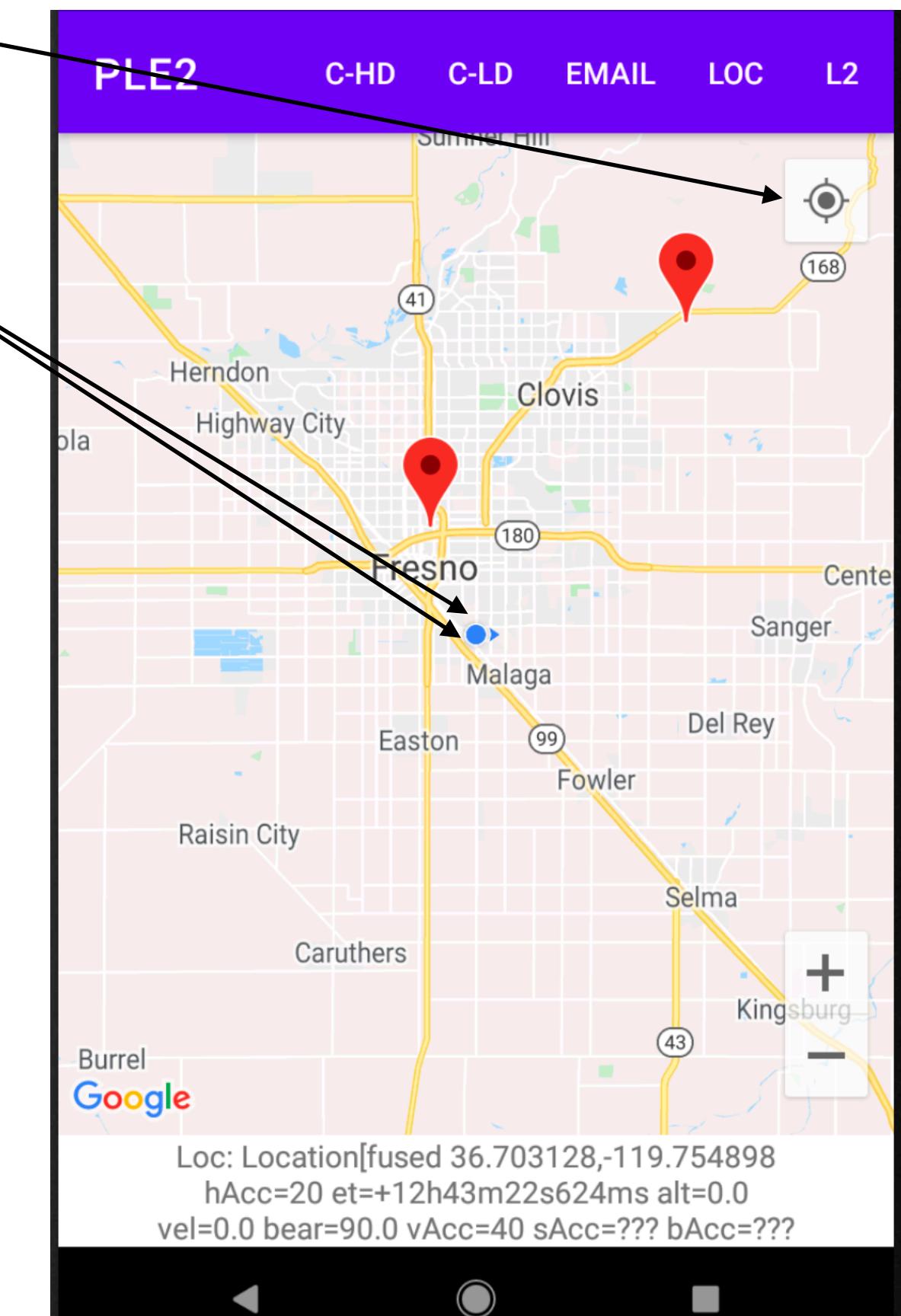
- Interact with another app
 - Intents
 - Go both ways
 - Your app can use other apps
 - Your app can be the “other app”
- Make calls to system components

More Map Code

- You can do a lot more with maps
 - Let people outline regions
 - etc

Could have had the fragment implement the interface (as with OnMyLocationClickListener) but chose to make an on the fly class

```
public class FragmentLocation extends Fragment implements  
OnMapReadyCallback, GoogleMap.OnMyLocationButtonClickListener,  
GoogleMap.OnMyLocationClickListener {  
  
    googleMap.setMyLocationEnabled(true);  
    googleMap.setOnMyLocationButtonClickListener(this);  
    googleMap.setOnMyLocationClickListener(this);  
  
    @Override  
    public boolean onMyLocationButtonClick() {  
        Log.i("THISS", "location button click ");  
        // do the default behavior which is zoom on current location  
        return false;  
    }  
  
    @SuppressLint("SetTextI18n")  
    @Override  
    public void onMyLocationClick(@NonNull Location location) {  
        Log.i("THISS", "Current location:\n" + location);  
        if (tv!=null)  
            tv.setText("Loc: " + location);  
        LatLng ll = new LatLng(location.getLatitude(),  
location.getLongitude());  
        zoomMap(ll);  
    }  
  
    googleMap.setOnMapClickListener(new  
    GoogleMap.OnMapClickListener() {  
        @Override  
        public void onMapClick(LatLng latLng) {  
            Log.i("THISS", "Clicked at " + latLng);  
            polyPoints.add(latLng);  
            Log.i("THISS", "BBBB" + polyPoints.size());  
            if (polyPoints.size()>3) {  
                PolygonOptions po = new PolygonOptions()  
                    .clickable(true);  
                for (LatLng ll : polyPoints)  
                    po.add(ll);  
                Polygon polygon1 = googleMap.addPolygon(po);  
                polygon1.setTag("alpha");  
                polygon1.setFillColor(Color.argb(128, 128,  
128, 0));  
                Log.i("THISS", "Added a polygon");  
                polyPoints.clear();  
            }  
        }  
    });  
}
```



Using Intents

- Point of intents is to allow application (specifically an activity) to get some other app to do something
 - two types
 - Just pass to other app/activity
 - Often intra-app, that is the app has more than one activity
 - Pass, but expect to get something back on completion
 - Take/get picture
 - Send Email
- PLE2: All intents started from MainActivity
 - They can start anywhere, but for this app they are from the actionBar so ...
 - Return comes to the activity (in onActivityResult)
 - if want otherwise need to write code
 - No effective way to passing values to yourself in intent other than the ID
 - The intent you get back is NOT the intent you sent
 - How to handle two different places in app requesting a picture?

Start intent and expect to get something back

The ID will come back in the OnActivityResult

Called when getting something back

Get the image

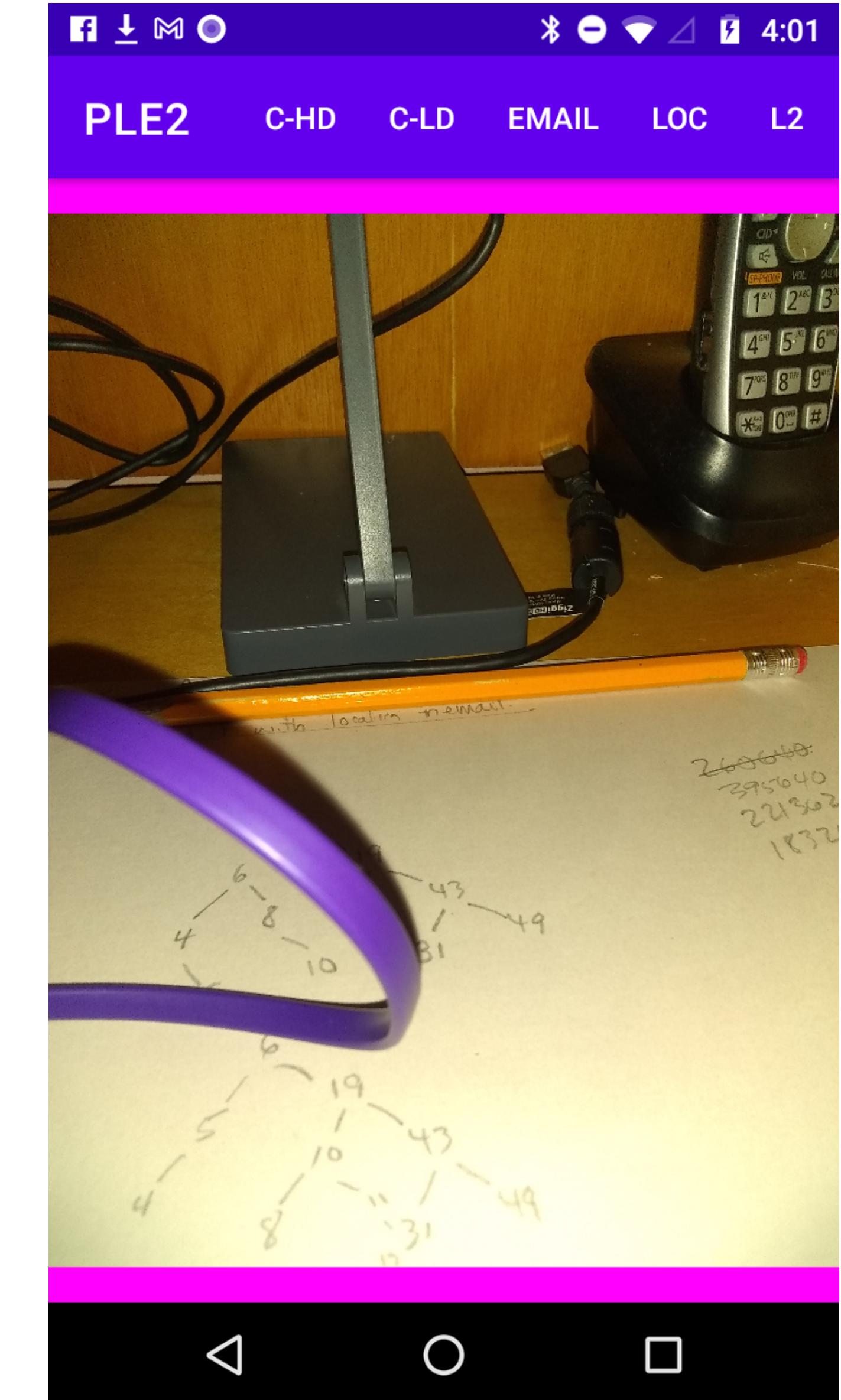
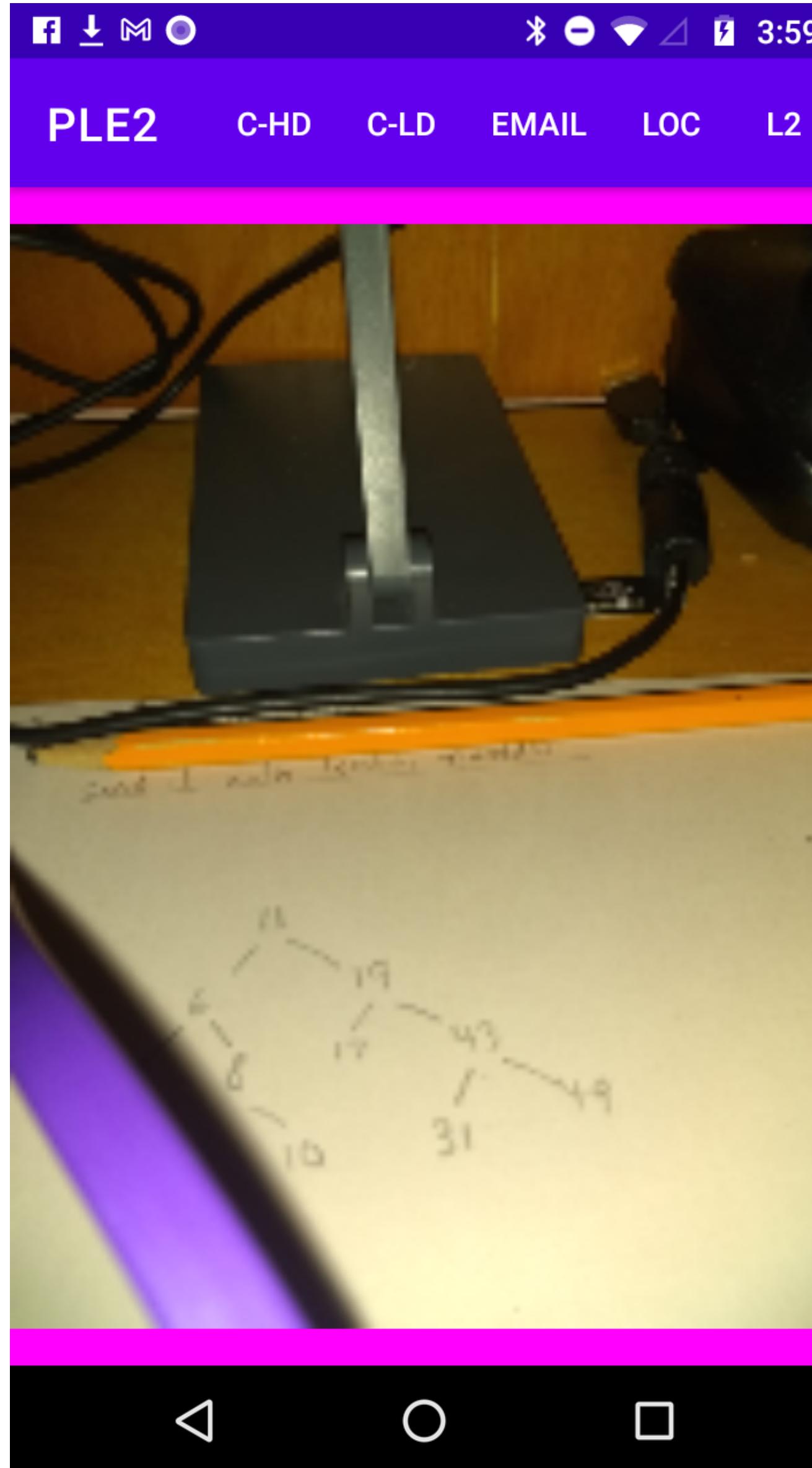
Show the image

Simple Photo Intent

```
public void doPhotoIntentSimple() {  
    Intent camera_intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
    startActivityForResult(camera_intent,  
        INTENT_ID_PIC_SIMPLE);  
}  
  
public void onActivityResult(int requestCode, int resultCode,  
    Intent intent ) {  
    super.onActivityResult(requestCode, resultCode, intent);  
    Log.i("THISS", "Receive intent " + intent);  
    if (requestCode==INTENT_ID_PIC_SIMPLE) {  
        if (Activity.RESULT_OK != resultCode) {  
            photoProblem();  
            return;  
        }  
  
        Bundle extras = intent.getExtras();  
        if (extras != null) {  
            Bitmap xx = (Bitmap) extras.get("data");  
            Log.i("THISS", "Bitmap " + xx.getByteCount() + "  
                " + xx.getWidth() + " " + xx.getHeight());  
            thePicture = xx;  
        }  
        switchToPhotoFragment();  
    }  
}
```

Photo Issues

- The simple photo system gets only thumbnails!
 - Hard to see, but image on left is only 153 x 204
 - Getting higher-res image takes more work!



Getting Hi-Res Images

- To get a hi-res image you need to pass to photo app a place to write the image.
- Android security model no longer has any shared file space
- Solution — FileProvider — new(ish) android annoyance.
 - Idea — do not give a file, give a writable URI. Since you opened the uri, this does not break security.
 - Problem: you have to remember that file so you can read it later

```
<!-- In AndroidManifest.xml -->
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="edu.brynmawr.ple2.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths"></meta-data>
</provider>

<!-- IN res/xml/file_paths.xml -->
<?xml version="1.0" encoding="utf-8"?>
<paths>
    <external-path
        name="external"
        path="." />
    <external-files-path
        name="external_files"
        path="." />
    <cache-path
        name="cache"
        path="." />
    <external-cache-path
        name="external_cache"
        path="." />
    <files-path
        name="files"
        path="." />
</paths>
```

```
private File photofile;

public void doPhotoIntent() {
    Log.e("THISSScF", "taking picture");
    Intent getIntent = new Intent(Intent.ACTION_GET_CONTENT);
    getIntent.setType("image/*");

    Intent takePictureIntent = new
    Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    try {
        photofile = createFileForImage();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    if (photofile != null) {
        Uri imageUri = FileProvider.getUriForFile(this,
            "edu.brynmawr.ple2.fileprovider",
            photofile);
        takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
        imageUri);
        Log.i("THISSS", "CALL INTENT " + takePictureIntent);
        startActivityForResult(takePictureIntent, INTENT_ID_PIC);
    }
}

public void onActivityResult(int requestCode, int resultCode,
Intent intent ) {
    super.onActivityResult(requestCode, resultCode, intent);
    if (requestCode==INTENT_ID_PIC)
    {
        if (Activity.RESULT_OK != resultCode) {
            photoProblem();
            return;
        }
        try
        {
            Bitmap bitmap =
            BitmapFactory.decodeFile(photofile.toString());
            thePicture=bitmap;
            photofile.delete();
            switchToPhotoFragment();
        }
        catch (Exception ee)
        {
            ee.printStackTrace();
            photoProblem();
        }
    }
}
```

Showing the Photo

- Use a new fragment that just shows an image
- Android ImageView widget
- Using static var – thePicture – on MainActivity kind of sucks
 - Would have been better to pass around the bitmap
 - But I wanted to use the image in other places and easiest way to do that is with a static variable
 - perhaps slightly better to put static in FragmentPhoto

```
// in MainActivity

private void switchToPhotoFragment() {
    FragmentManager fragmentManager = this.getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    transaction.replace(R.id.mainlinearLayout, new FragmentPhoto(), null);
    transaction.addToBackStack(null);
    transaction.commit();
}

//In FragmentPhoto
public void onViewCreated(@NotNull View view, Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    layoutDoPhoto();
}

private void layoutDoPhoto() {
    mainView.removeAllViews();
    mainView.setBackgroundColor(Color.MAGENTA);
    ImageView iv = new ImageView(getContext());
    iv.setScaleType(ImageView.ScaleType.FIT_CENTER);
    iv.setImageBitmap(MainActivity.thePicture);
    mainView.addView(iv, new RelativeLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, vie
```

Sending Email

Another Intent

- Sending a photo in email
 - Need to do the same FileProvider thing
 - This time to give a readable link
 - Since the image file was deleted (and may never have been created) make an image file
 - Also allows for format choice
 - jpeg, png, webp
 - webp claims better compression with same loss
 - But not universally readable (now) so using jpeg
 - The email intent does not return a meaningful result code

```
private void doEmail() {
    try {
        String s="Default text";
        if (currentLocation!=null)
            s += String.format("Your current location: %6.2f, %6.2f",
currentLocation.getLongitude(), currentLocation.getLatitude());
        Intent i = new Intent(Intent.ACTION_SEND);
        i.setType("application/image");
        i.putExtra(Intent.EXTRA_SUBJECT, "The photo I just took");
        i.putExtra(Intent.EXTRA_TEXT, s);
        File ff = createFileContainingImage();
        if (ff!=null) {
            Log.i("THIS", "PHOTO URI: " + ff.length() + " " + ff.toURI());
            Uri imageUri = FileProvider.getUriForFile(this,
                "edu.brynmawr.ple2.fileprovider",
                ff);
            i.putExtra(Intent.EXTRA_STREAM, imageUri);
        }
        this.startActivityForResult(Intent.createChooser(i, "Send mail..."),
INTENT_ID_EMAIL);
    }
    catch (Exception ee)
    {
        ee.printStackTrace();
    }
}

public void onActivityResult(int requestCode, int resultCode, Intent intent )
{
    super.onActivityResult(requestCode, resultCode, intent);
    if (requestCode == INTENT_ID_EMAIL) {
        Log.i("THIS", "EMail intent completed ... result is unknown and unknowable per
spec");
        return;
    }

    private File createFileContainingImage() throws IOException {
        // Create a file
        if (thePicture==null) return null;
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
        String imageFileName = "JPEG_" + timeStamp + "_";
        File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
        File image = File.createTempFile( imageFileName, ".jpg", storageDir );
        try {
            FileOutputStream out = new FileOutputStream(image);
            thePicture.compress(Bitmap.CompressFormat.JPEG, 90, out);
            out.flush();
            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return image;
    }
}
```

Receiving Intents

- Passed from activity to activity within an app
 - passed info must be serializable or parcelable
 - Equivalent, but parcelable is only android and 10x faster
- Passed from another app

```
// In MainActivity.java
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    Log.i("THIS", "Clicked on menu item " + id);

    if (id==MAT_SECOND) {
        Intent intent = new Intent(MainActivity.this,
                                    SecondActivity.class);
        intent.putExtra("username", "UserName");
        intent.putExtra("password", "UserPassword");
        startActivity(intent);
    }
    // ...

// In SecondActivity.java
public void onStart() {
    super.onStart();
    String value = getIntent().getStringExtra("username");
    String pass_val = getIntent().getStringExtra("password");
    // ...
}
```

Source Activity

Target Activity

Info to pass

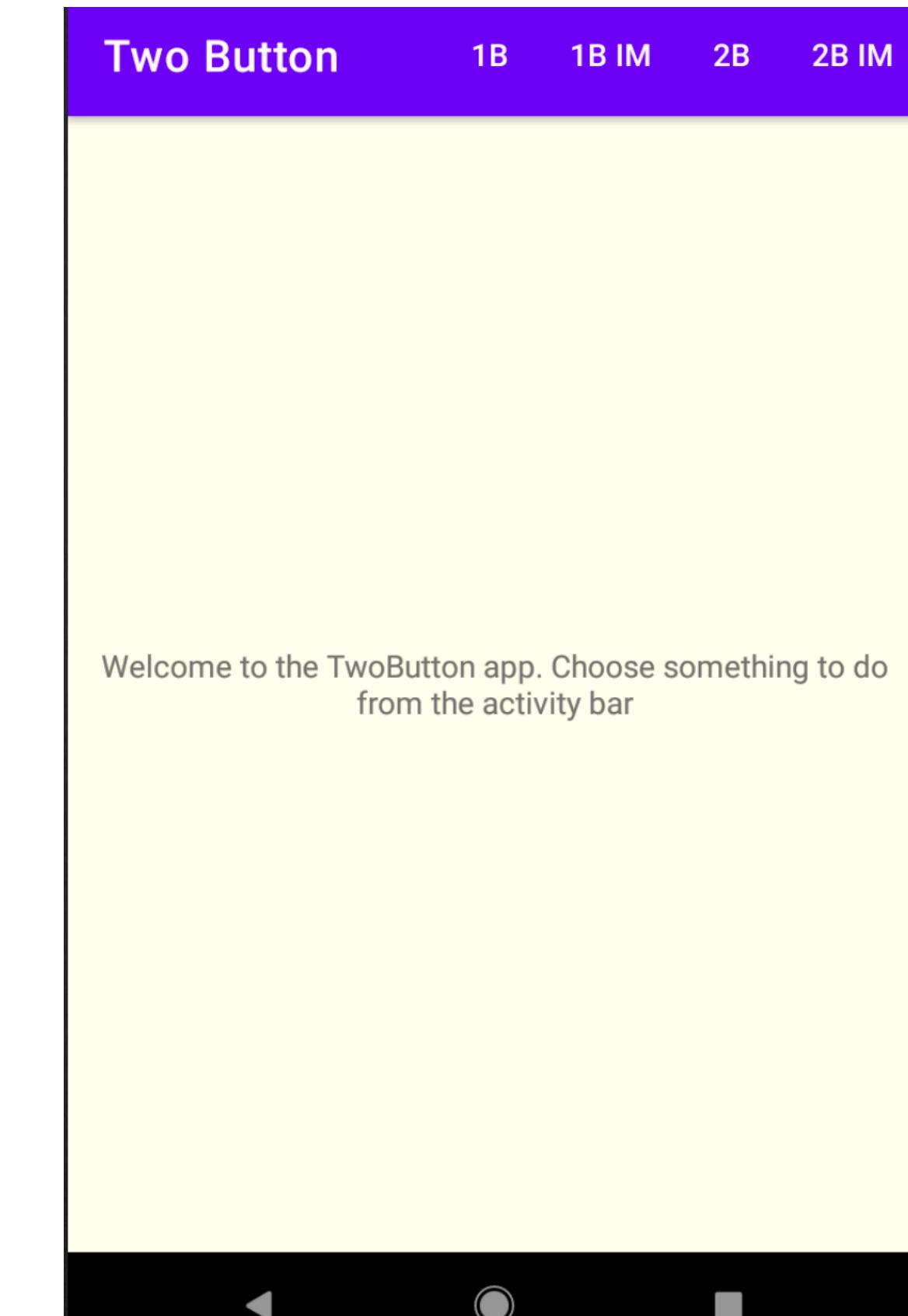
```
<!-- In AndroidManifest.xml -->
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="android.intent.action.EDIT" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data
        android:mimeType="text/plain"
    />
</intent-filter>
```

Implementing Interfaces

vs

Creating a receiver as needed

- The TwoButton app
- Same basic interface as PLE2, but does a whole lot less
- Just illustrates receiving the click of one or two buttons



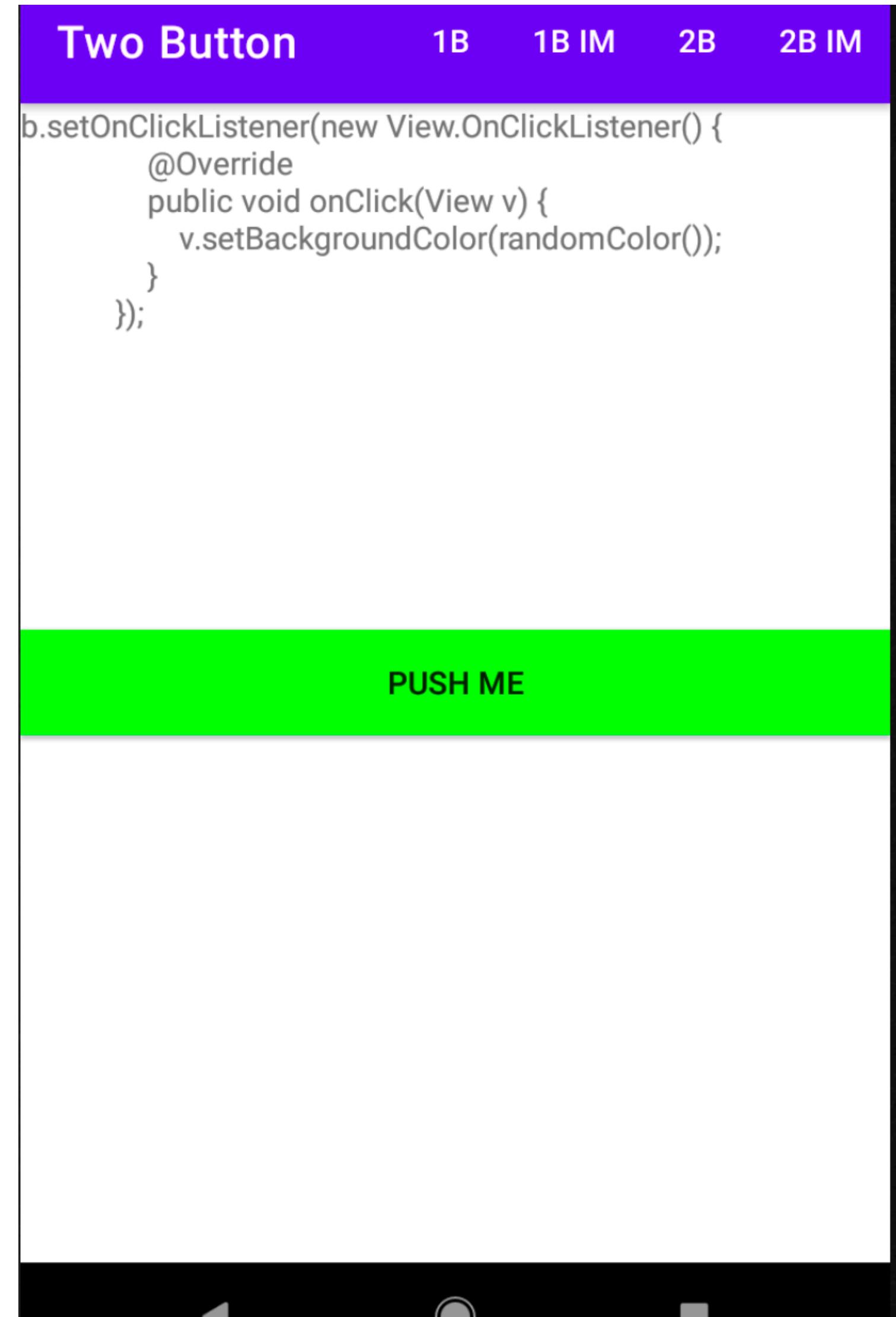
One Button

- Choice 1: implement the `View.OnClickListener` Interface

```
public class FragmentOneButtonIM extends Fragment implements  
View.OnClickListener{  
  
    // ...  
    Button b = new Button(getContext());  
    b.setOnClickListener(this);  
  
    // ...  
    @Override  
    public void onClick(View v) {  
        v.setBackgroundColor(randomColor());  
    }  
}
```

- Choice 2: create a class that implements `View.OnClickListener` exactly when and where needed

```
Button b = new Button(getContext());  
b.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        v.setBackgroundColor(randomColor());  
    }  
});
```



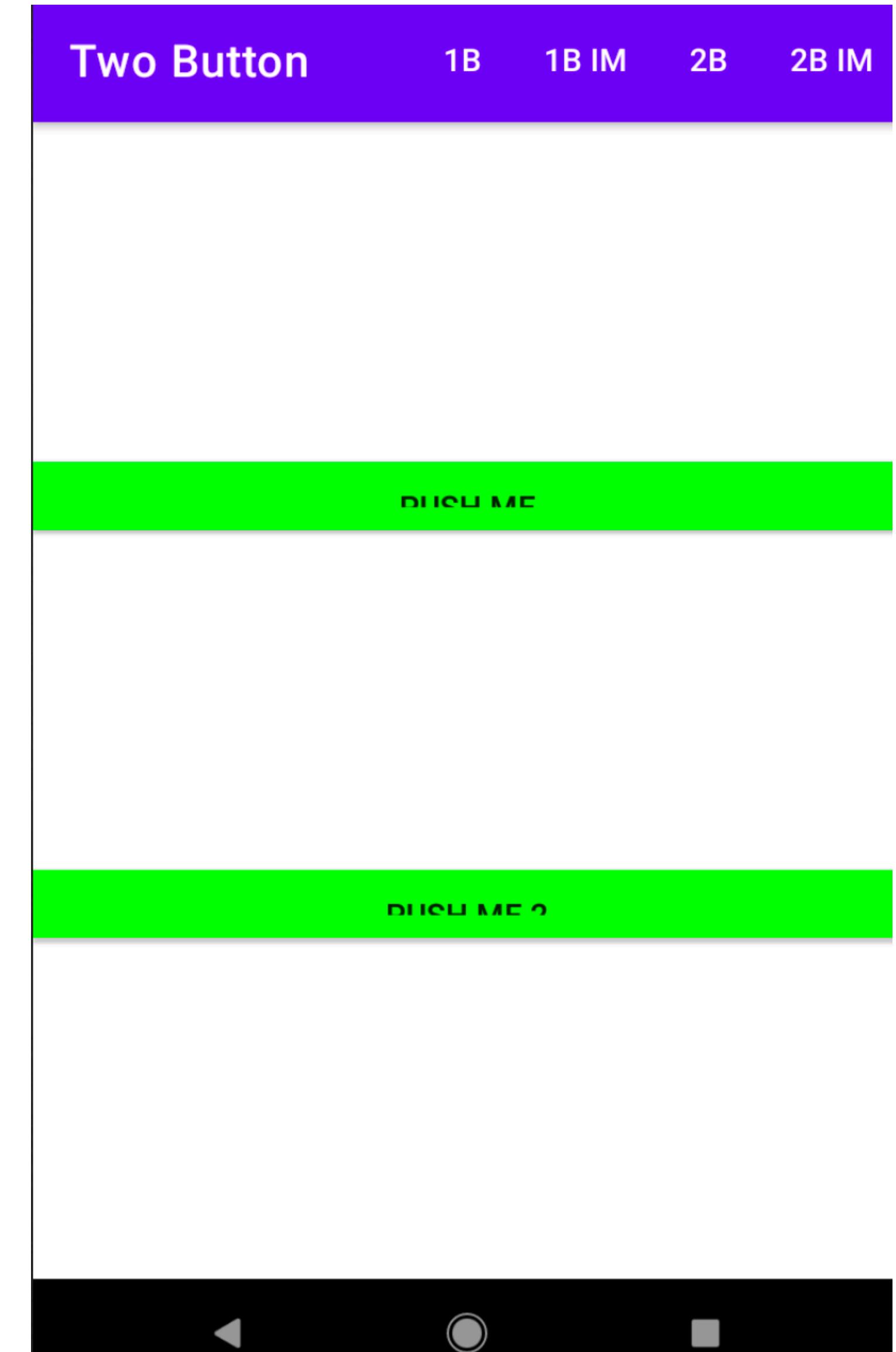
Two Buttons

- Choice 1: implement the `View.OnClickListener` Interface

```
public class FragmentOneButtonIM extends Fragment implements  
View.OnClickListener{  
  
// ...  
Button b1 = new Button(getContext());  
b1.setId(1)  
b1.setOnClickListener(this);  
Button b2 = new Button(getContext());  
b2.setId(2)  
b2.setOnClickListener(this);  
  
// ...  
@Override  
public void onClick(View v) {  
    if (v.getId()==1)  
}
```

- Choice 2: implements `onClickListener` as needed

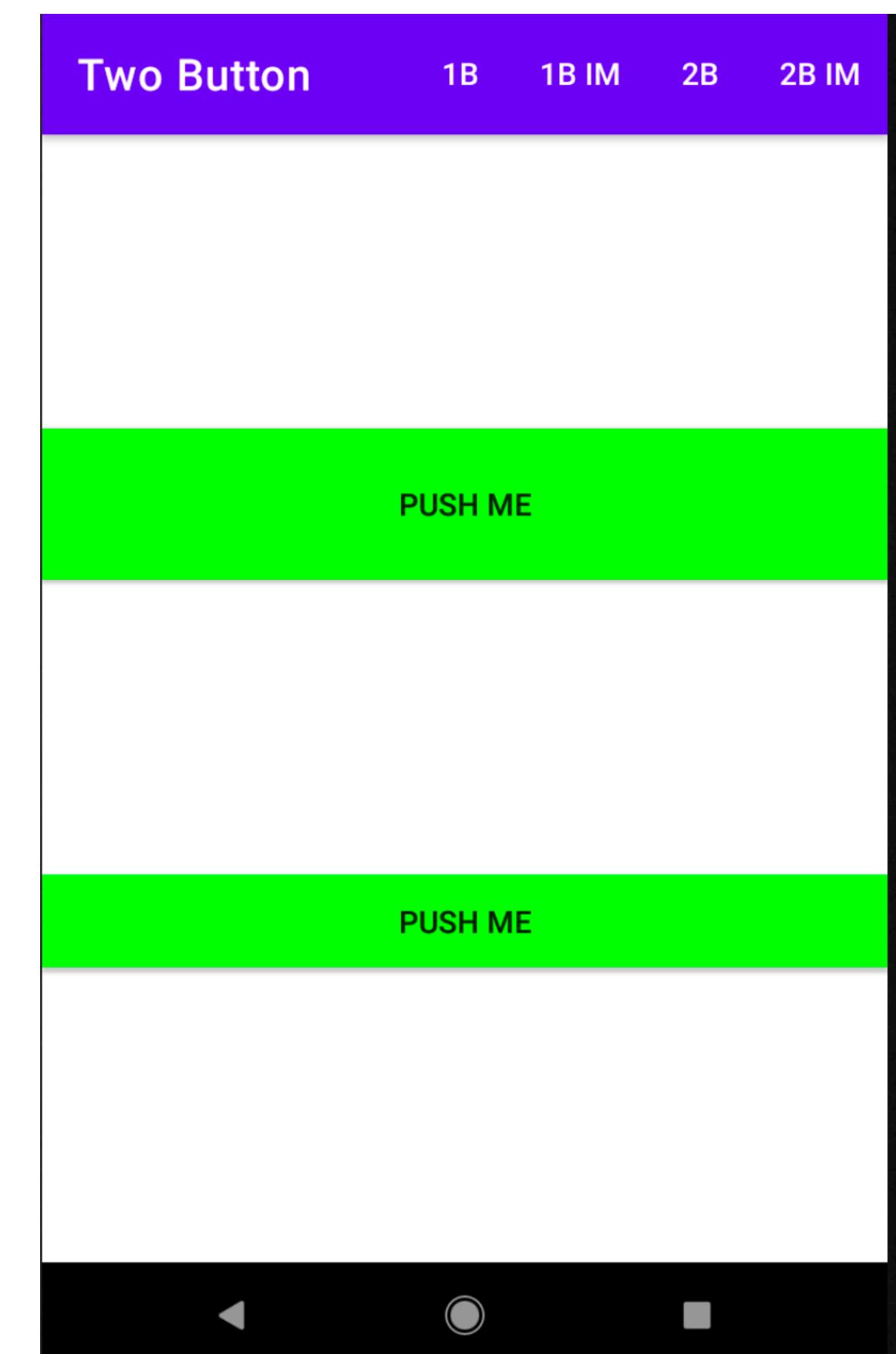
```
Button b1 = new Button(getContext());  
b1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        v.setBackgroundColor(randomColor());  
    }  
});  
Button b2 = new Button(getContext());  
b2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        v.setBackgroundColor(randomColor());  
    }  
});
```



LinearLayouts

- Do not just use weights
do allocate space.
- Especially for
buttons it is far better
to explicitly allocate
space
- Android (or Apple)
recommends 42dp
(minimum)

```
private void layoutDoButton() {
    float density =
        getContext().getResources().getDisplayMetrics().density;
    mainLayout.removeAllViews();
    mainLayout.addView(new View(getContext()), new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
    0, 5));
    {
        Button b = new Button(getContext());
        b.setText("Push me");
        b.setId(FIRST_BUTTON);
        b.setBackgroundColor(Color.GREEN);
        mainLayout.addView(b, new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
    (int)(42*density), 1));
        b.setOnClickListener(this);
    }
    mainLayout.addView(new View(getContext()), new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
    0, 5));
    {
        Button b = new Button(getContext());
        b.setText("Push me");
        b.setId(SECOND_BUTTON);
        b.setBackgroundColor(Color.GREEN);
        mainLayout.addView(b, new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
    (int)(42*density), 0));
        b.setOnClickListener(this);
    }
    mainLayout.addView(new View(getContext()), new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
    0, 5));
}
```



Scope, Buttons and Timers in Android

- Java/Android

- Recall that changes to UI MUST be done from UI thread.
 - onClick runs in UI thread
- variables used in delayed functions must be in scope (this includes onClick)
 - instance variable of containing class
 - e.g. pval
 - (effectively) final in method
 - e.g. tv1
 - These are somewhat safer
 - fewer possible null pointer exceptions

```
String pval="AAA";
int ival=0;
private void layoutDoButton() {
    mainLayout.removeAllViews();
    TextView tv1 = new TextView(getContext());
    mainLayout.addView(tv1, new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 5));
    {
        Button b = new Button(getContext());
        b.setText("Push me");
        b.setBackgroundColor(Color.GREEN);
        mainLayout.addView(b, new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 1));
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Handler handler = new Handler(Looper.getMainLooper());
                handler.postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        tv1.setText(pval.substring(pval.length()>0?pval.length()-1:0)+(ival++));
                    }
                }, 2000);
            }
        });
    }
    mainLayout.addView(new View(getContext()), new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 1));
    {
        Button b = new Button(getContext());
        b.setText("Push me 2");
        b.setBackgroundColor(Color.GREEN);
        mainLayout.addView(b, new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 1));
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mainLayout.setBackgroundColor(randomColor());
                pval=null;
            }
        });
    }
    mainLayout.addView(new View(getContext()), new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 1));
}
```

Scope, Timers and Javascript

- Scoping is much the same in Javascript timers as Java
 - All global scoped vars
 - All locals vars in current scope
 - Note difference: need not be final
- What is the output of the javascript?

```
const thediv = $("#output");
function dotest() {
    thediv.html("");
    let ii=0;
    while (ii<4) {
        thediv.html(thediv.html() + "<br>AA" + ii);
        setTimeout(function() {
            thediv.html(thediv.html() + "<br>BB " + ii );
        }, 1000);
        ii++;
    }
    ii+=1000;
}
```

Javascript and Timers

fixed?

- perhaps the problem is with the anonymous function?
 - so define a new function in the global function name space
- Output?

```
function dotest3() {  
    thediv.html("");  
    for (ii=0; ii<4; ii++) {  
        thediv.html(thediv.html() + "<br>DD " + ii);  
        setTimeout(function() { doFunc(ii); }, 1000);  
    }  
    ii+=3000;  
}  
function doFunc(i3) {  
    thediv.html(thediv.html() + "<br>FUNC " + i3 );  
}
```

Javascript and Timers

Resolved

- var variables are “function scoped” and redefinable.
 - So every time var is used within loop it overwrites old value
- let,const are block scoped and NOT redefinable.
 - So every time they are used within loop they are created anew
- There are other ways to do this. I like the “declaring const variables just before timeout” approach. It makes things obvious.

```
function dotest5() {  
    thediv.html("");  
    noww = Date.now();  
    for (ii=0; ii<4; ii++) {  
        thediv.html(thediv.html() + "<br>T5 " + ii);  
        const ii5=ii;  
        setTimeout(function() { doFunc(ii5); } , 1000);  
    }  
    ii+=5000;  
}
```