

Android System Tools

Tool Types

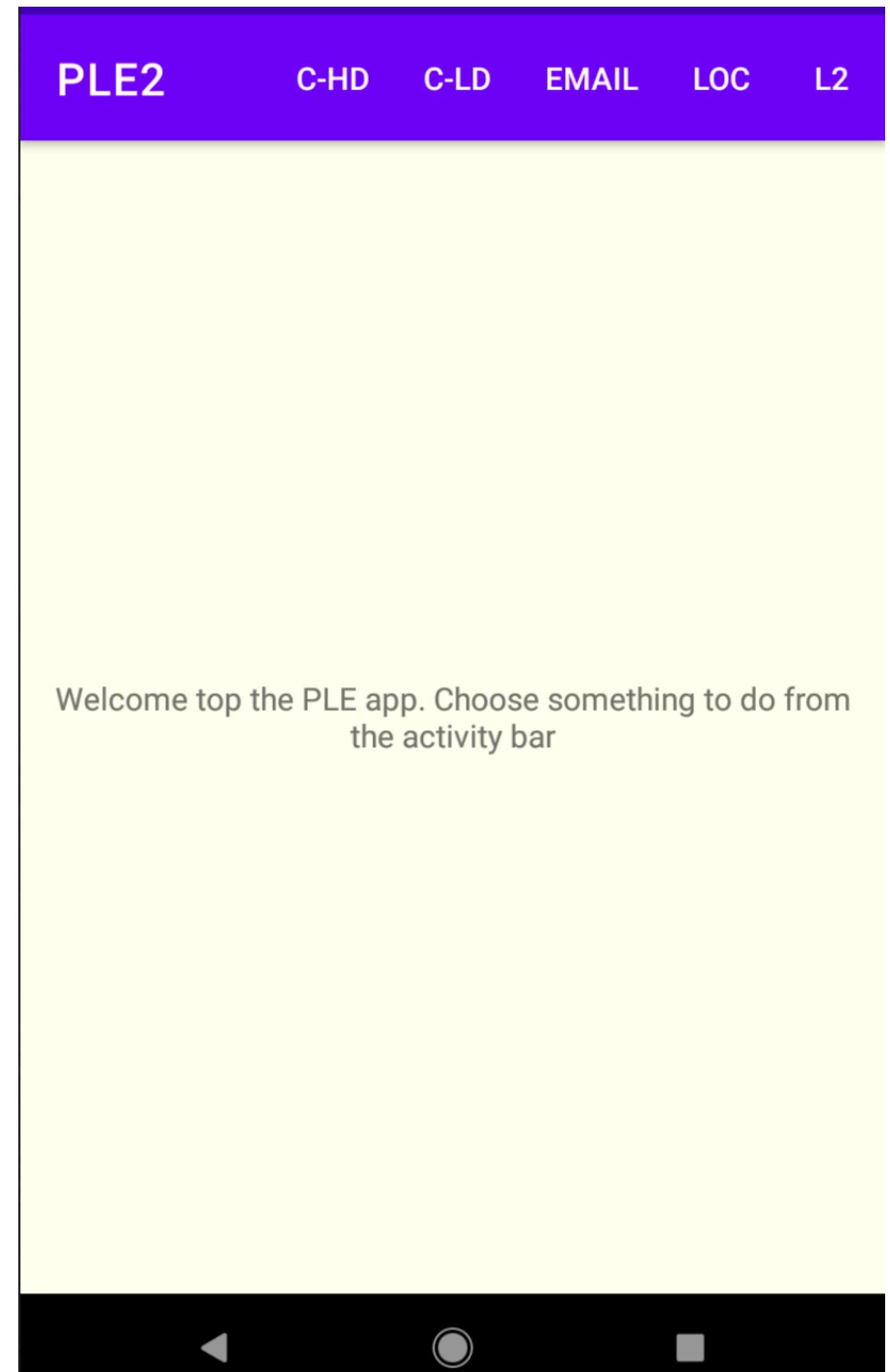
- Interact with another app
 - Intents
 - Go both ways
 - Your app can use other apps
 - Your app can be the “other app”
- Make calls to system components

Intents

- Asynchronous messages which
 - allow functionality requests from other Android components.
 - from the same application
 - From other applications.
 - E.g. taking a picture.
 - Send an email

My app for today

- Action bar to select actions
- Fragments to show results of selections
- All navigation and menu bar in code
- Actions:
 - take pictures
 - send email
 - show maps
 - show distances
- Name: PLE2 “Pictures, Location, Email v2”



Starting Code

- Android Manifest
 - specifies starting Activity
- theme.xml
 - sets up some basic params of app
 - ActionBar
- activity_main.xml
 - Describes the base layout manager for app
 - Nothing is in the activity layout
- MainActivity.java
 - code defining the activity

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.PLE2" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <item name="android:statusBarColor" tools:targetApi="1">?attr/colorPrimaryVariant</item>
    </style>
</resources>
```

change to
NoActionBar

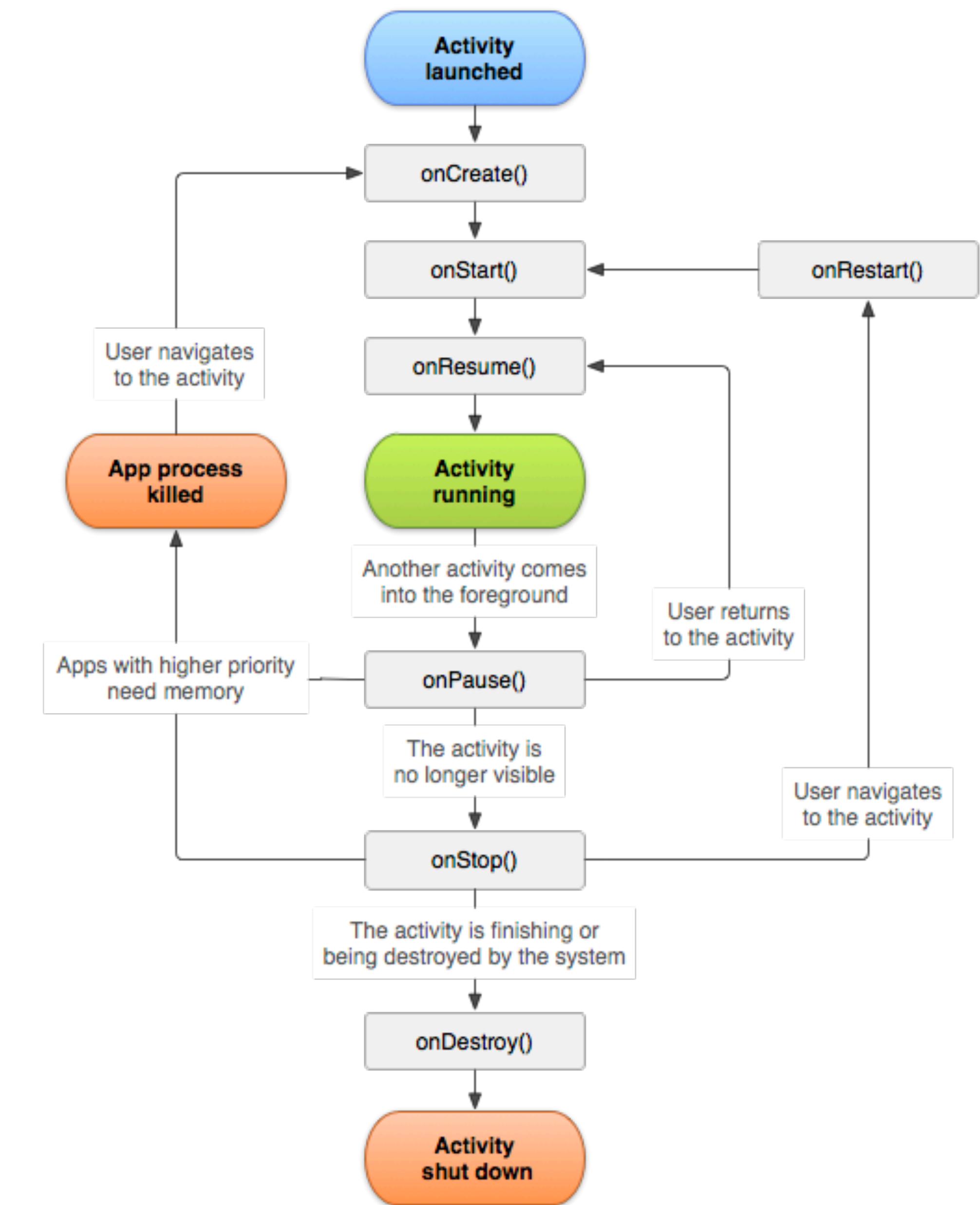
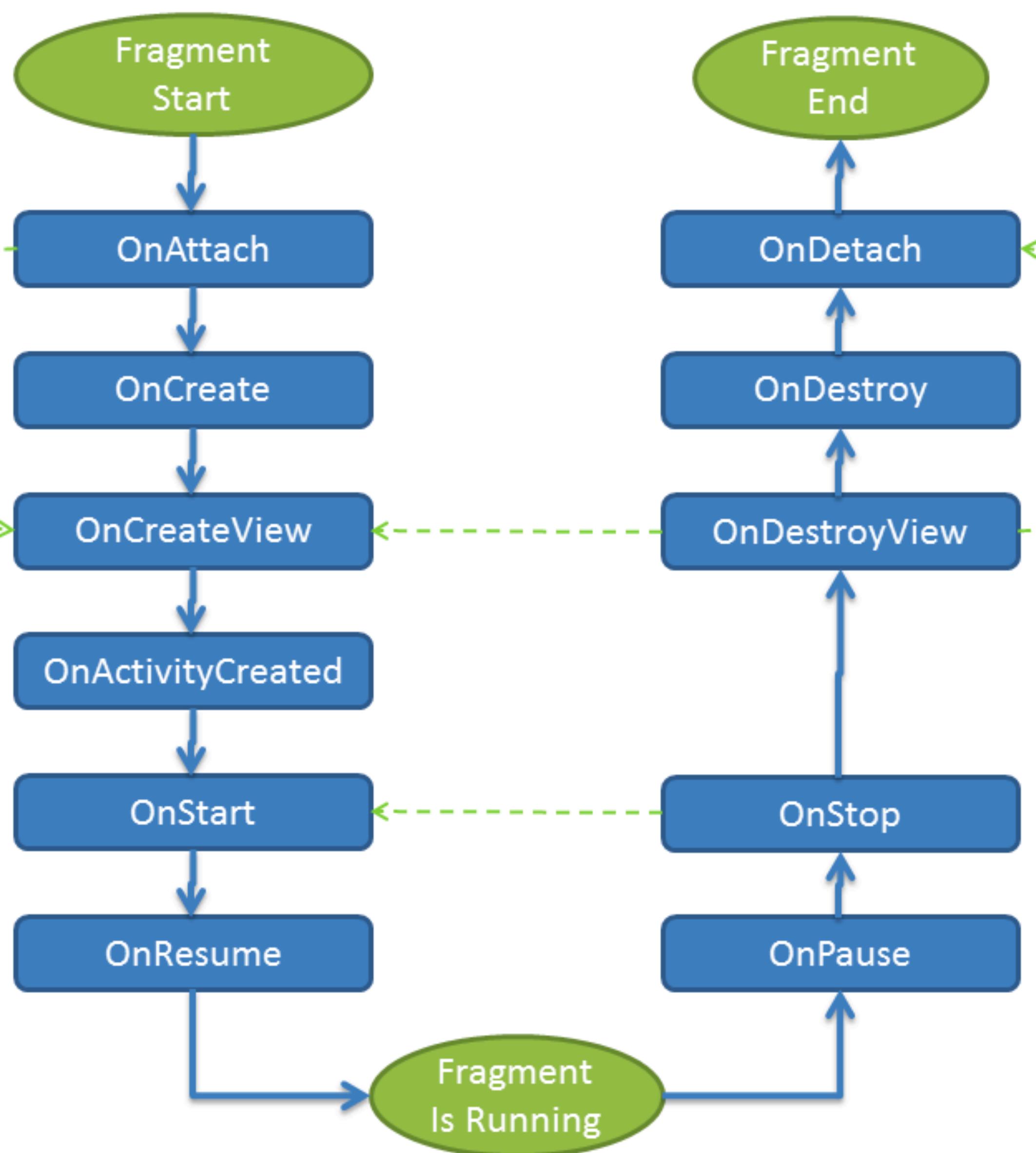
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/mainlinearlayout"
    tools:context=".MainActivity" >
</RelativeLayout>
```

Specify that the activity should first display the contents of FragmentWelcome

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    FragmentManager fragmentManager = this.getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    transaction.replace(R.id.mainlinearlayout, new FragmentWelcome(), null);
    transaction.addToBackStack(null);
    transaction.commit();
    int tt = fragmentManager.getBackStackEntryCount();
    for(int i = 0; i < tt; ++i) {
        fragmentManager.popBackStack();
    }
}
```

Do not allow the Android back button to back out of the app

Fragment Lifecycles Activity



Actions in the MainActivity

- ALL of this is event driven
- All of this comes here regardless of where in app the click occurs
- Action Bar
 - Create the contents of the bar
 - Respond to clicks in the bar
- Handle Android Back Button

All of the "MAT_" vars are constants defined in the class

This keeps the back button from leaving the app.
But allows the it to navigate back in fragments

```
public boolean onCreateOptionsMenu(Menu menu){  
    MenuItem edit_item = menu.add(0, MAT_CAMERA, 1, "C-HD");  
    edit_item.setShowAsActionFlags(MenuItem.SHOW_AS_ACTION_ALWAYS); }  
  
    MenuItem edit_item = menu.add(0, MAT_CAMERA2, 1, "C-LD");  
    edit_item.setShowAsActionFlags(MenuItem.SHOW_AS_ACTION_ALWAYS); }  
  
    MenuItem edit_item = menu.add(0, MAT_EMAIL, 1, "EMAIL");  
    edit_item.setShowAsActionFlags(MenuItem.SHOW_AS_ACTION_ALWAYS); }  
  
    MenuItem edit_item = menu.add(0, MAT_LOCATE, 1, "Loc");  
    edit_item.setShowAsActionFlags(MenuItem.SHOW_AS_ACTION_ALWAYS); }  
  
    MenuItem edit_item = menu.add(0, MAT_LOCATE2, 1, "L2");  
    edit_item.setShowAsActionFlags(MenuItem.SHOW_AS_ACTION_ALWAYS); }  
  
}  
  
public boolean onOptionsItemSelected(MenuItem item) {  
    int id = item.getItemId();  
    if (id==MAT_CAMERA) {  
        doPhotoIntent();  
        return true; }  
    if (id==MAT_CAMERA2) {  
        doPhotoIntentSimple();  
        return true; }  
    if (id == MAT_EMAIL) {  
        doEmail2();  
        return true; }  
    if (id == MAT_LOCATE) {  
        Toast.makeText(this, "Location", Toast.LENGTH_LONG).show();  
        FragmentManager fragmentManager = this.getSupportFragmentManager();  
        FragmentTransaction transaction = fragmentManager.beginTransaction();  
        transaction.replace(R.id.mainlinearlayout, new FragmentLocation(), null);  
        transaction.addToBackStack(null);  
        transaction.commit();  
        return true; }  
    if (id == MAT_LOCATE2) {  
        // LIKE MAT_LOCATE  
    }  
    return super.onOptionsItemSelected(item); }  
  
public void onBackPressed() {  
    FragmentManager fragmentManager = this.getSupportFragmentManager();  
    int tt = fragmentManager.getBackStackEntryCount();  
    Log.i("THISs", "Back stack " + tt);  
    if (tt>1)  
        super.onBackPressed(); }  
}
```

Change the fragment being displayed

Fragment Locate2

Show distance to cities in US

- Make call to system to get GPS coordinates
 - Asynchronous
- Show distances in list view

PLE2	C-HD	C-LD	EMAIL	LOC	L2
Current Location: Lat: 36.70 Long:-119.75					
Fresno, California					
Providence Rhode Island				4145.80 km	
Warwick Rhode Island				4146.98 km	
Pawtucket Rhode Island				4148.25 km	
Lawrence Massachusetts				4152.47 km	
Newton Massachusetts				4154.79 km	
Cambridge Massachusetts				4161.63 km	
Somerville Massachusetts				4162.00 km	
Boston Massachusetts				4165.90 km	
Lynn Massachusetts				4172.14 km	
Quincy Massachusetts				4172.17 km	
Fall River Massachusetts				4173.07 km	
Brockton Massachusetts				4173.43 km	
New Bedford Massachusetts				4187.88 km	

List Views

- 2 Parts

- the ListView object

- specifies general properties

- The “Adapter”

- Specifies the list to be shown

- How to display items in the list

- how to respond to click on list item

- Most commonly an “ArrayAdapter”

```
ListView theLv = new ListView(getContext());
mainView.addView(theLv, new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 9));
theLv.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
theLv.setBackgroundColor(Color.rgb(240, 240, 240));
theLv.setDivider(new ColorDrawable(0xff8B008B));
theLv.setDividerHeight((int)(2*getResources().getDisplayMetrics().density));
theLv.setStackFromBottom(true);
theLv.setTranscriptMode(ListView.TRANSCRIPT_MODE_ALWAYS_SCROLL);
cityAdapter = new ArrayAdapter<City>(getContext(), 0, DataCities.getInstance().getCities())
{
    public View getView(int position, View convertView, @NonNull ViewGroup parent)
    {
        View view = convertView;
        if (view == null)
        {
            view = new LinearLayout(getContext());
        }
        City msg = this.getItem(position);
        LinearLayout ll = (LinearLayout) view;
        ll.removeAllViews();
        ll.setOrientation(LinearLayout.HORIZONTAL);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        {
            TextView tv = new TextView(getContext());
            tv.setText(msg.name + " " + msg.state);
            tv.setGravity(Gravity.LEFT | Gravity.CENTER_VERTICAL);
            tv.setPadding((int)(3*density), (int)(8*density), 0, (int)(8*density));
            ll.addView(tv, new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.WRAP_CONTENT));
        }
        {
            TextView tv = new TextView(getContext());
            tv.setText(String.format("%8.2f km", msg.getDistance()/1000));
            tv.setGravity(Gravity.LEFT | Gravity.CENTER_VERTICAL);
            float density = getResources().getDisplayMetrics().density;
            tv.setPadding((int)(3*density), (int)(8*density), 0, (int)(8*density));
            ll.addView(tv, new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.WRAP_CONTENT));
        }
        return view;
    }
};
theLv.setAdapter(cityAdapter);
}
```

Getting GPS coords

in build.gradle add some imports

```
implementation 'com.google.android.gms:play-services-location:17.0.0'  
implementation 'com.google.android.gms:play-services-maps:17.0.0'  
implementation 'com.google.maps.android:android-maps-utils:0.5'
```

- Create a LocationManager
- Set up a location “listener”
 - this receives GPS coordinates
 - When location has changed update all of the distances, then tell the list adapter the data has changed
 - Get the town name to show user
 - Make sure that the user has allowed the app to use GPS

```
LocationManager locationManager = (LocationManager)  
    getContext().getSystemService(Context.LOCATION_SERVICE);  
  
LocationListener locationListener = new MyLocationListener();  
if (ActivityCompat.checkSelfPermission(getContext(),  
    Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&  
    ActivityCompat.checkSelfPermission(getContext(), Manifest.permission.ACCESS_COARSE_LOCATION)  
    != PackageManager.PERMISSION_GRANTED) {  
    if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),  
        Manifest.permission.ACCESS_FINE_LOCATION)) {  
        Toast.makeText(getContext(), "The app needs to access your location to do things  
        with the map", Toast.LENGTH_LONG).show();  
    } else {  
        // No explanation needed; request the permission  
        ActivityCompat.requestPermissions(getActivity(),  
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},  
            1476);  
    }  
}  
locationManager.requestLocationUpdates(  
    LocationManager.GPS_PROVIDER, 20, 10, locationListener);  
  
private class MyLocationListener implements LocationListener {  
    @Override  
    public void onLocationChanged(Location loc) {  
        MainActivity.currentLocation=loc;  
        String cityName = null;  
        Geocoder gcd = new Geocoder(getApplicationContext(), Locale.getDefault());  
        List<Address> addresses;  
        try {  
            addresses = gcd.getFromLocation(loc.getLatitude(), loc.getLongitude(), 1);  
            if (addresses.size() > 0) {  
                cityName = addresses.get(0).getLocality();  
            }  
            currentLocView.setText(String.format("Current Location: Lat:%.6.2f Long:  
%.6.2f\n%s",  
                loc.getLatitude(), loc.getLongitude(), cityName,  
                addresses.get(0).getAdminArea()));  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        LatLng ll = new LatLng(loc.getLatitude(), loc.getLongitude());  
        for (City c : DataCities.getInstance().getCities()) {  
            c.setDistanceFrom(ll);  
        }  
        DataCities.getInstance().sortByDistance();  
        cityAdapter.notifyDataSetChanged();  
    }  
}
```

Showing Maps

- Requires the same build.gradle additions from distance
- Also needs “key” from google in manifest
- Initialize GPS
- Set up the display (some of this could be in XML)
- Init maps and set up Fragment as a map listener (it implements an interface)
- When the maps are ready
 - set up some more listeners
 - move the map to my location
 - Put down markers for cities in my list

```
public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    LocationManager locationManager = (LocationManager)
        getApplicationContext().getSystemService(Context.LOCATION_SERVICE);
    // code to confirm permission not shown – same as before
    locationManager.requestLocationUpdates(
        LocationManager.GPS_PROVIDER, 20, 10, locationListener);
    layoutMap(savedInstanceState);
}

private void layoutMap(Bundle savedInstanceState) {
    mapView = new MapView(getApplicationContext());
    mainView.addView(mapView, new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 9));
    mapView.onCreate(savedInstanceState);
    mapView.onResume(); // needed to get the map to display immediately

    try { MapsInitializer.initialize(getActivity().getApplicationContext());
    } catch (Exception e) { e.printStackTrace(); }
    mapView.getMapAsync(this);

    tv = new TextView(getApplicationContext());
    tv.setText("Location!!!");
    tv.setGravity(Gravity.CENTER);
    tv.setPadding((int)(10*getResources().getDisplayMetrics().density), 0, (int)(10*getResources().getDisplayMetrics().density), 0);
    LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0);
    mainView.addView(tv, lp);
}

@Override
public void onMapReady(GoogleMap googleMap) {
    try {
        googleMap.setMyLocationEnabled(true);
        googleMap.setOnMyLocationButtonClickListener(this);
        googleMap.setOnMyLocationClickListener(this);
    } catch (Exception ee) {
        Log.e("KIKL", "Problematic " + ee.toString());
        ee.printStackTrace();
    }
    googleMap.getUiSettings().setZoomControlsEnabled(true);
    LatLng mCurrentLocation = getLocation();
    if (mCurrentLocation==null)
        mCurrentLocation = new LatLng(39.92, -75.63);
    this.googleMap=googleMap;
    zoomMap(mCurrentLocation);

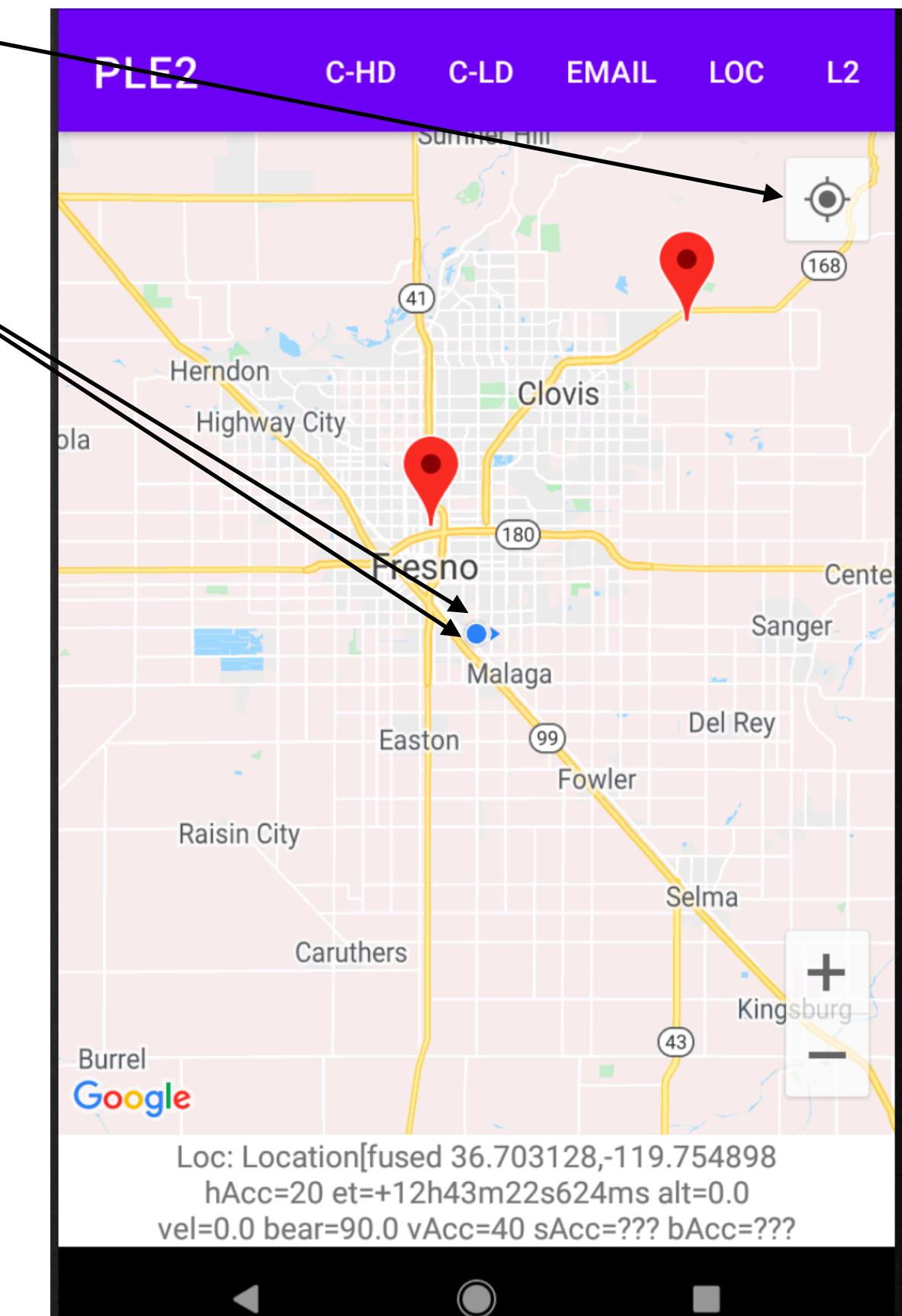
    for (City c : DataCities.getInstance().getCities()) {
        MarkerOptions markerOptions = new MarkerOptions();
        markerOptions.position(new LatLng(c.getLatitude(), c.getLongitude()));
        markerOptions.title(c.name);
        markerOptions.snippet(c.state);
        Marker m = googleMap.addMarker(markerOptions);
        m.setTag("PIN");
    }
}
```

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
 - 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 ...
 - If getting something back, that return does come to the activity

Start intent and expect to get something back

The ID will come back in the OnActivityResult

Called when getting something back

Get the image

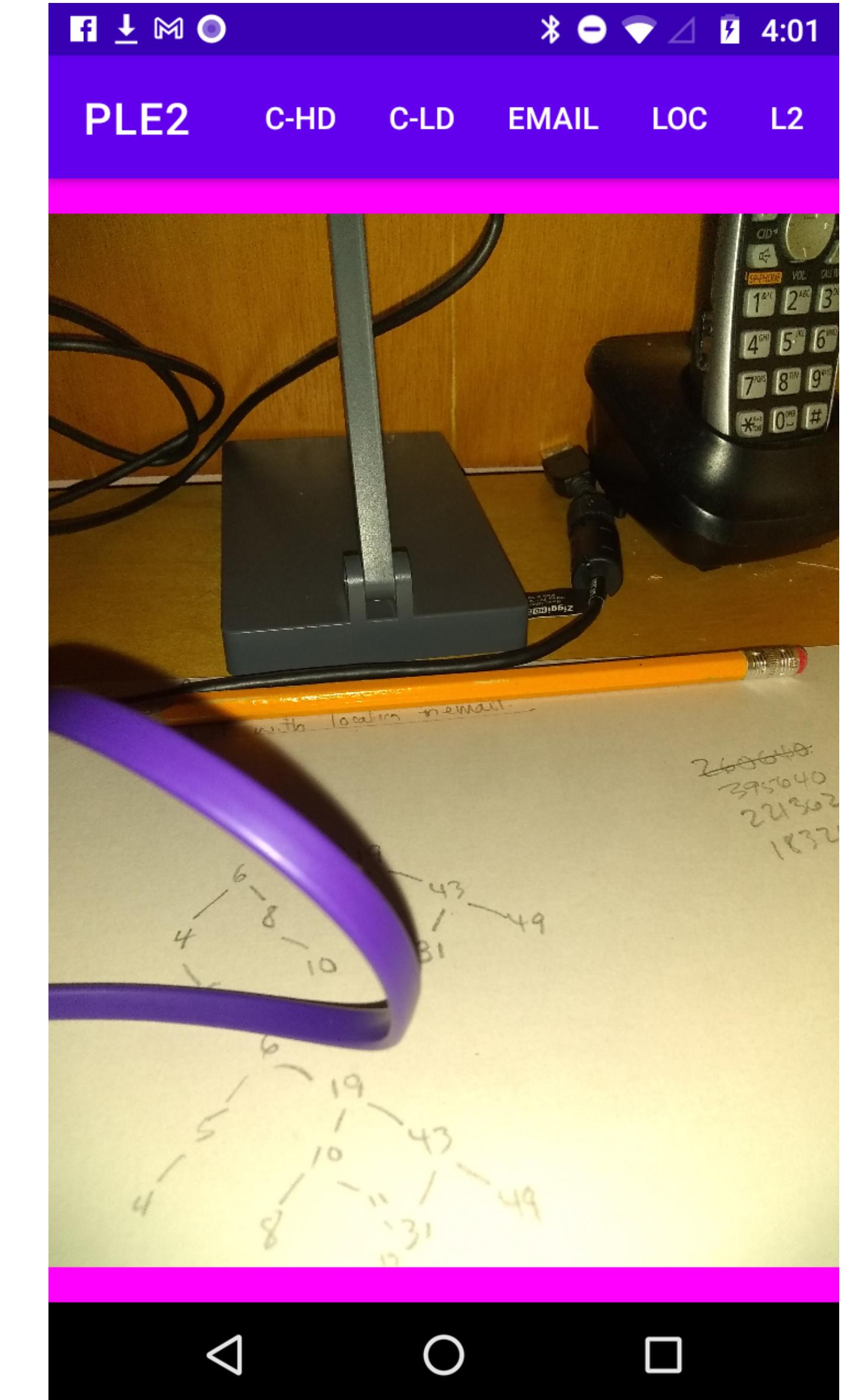
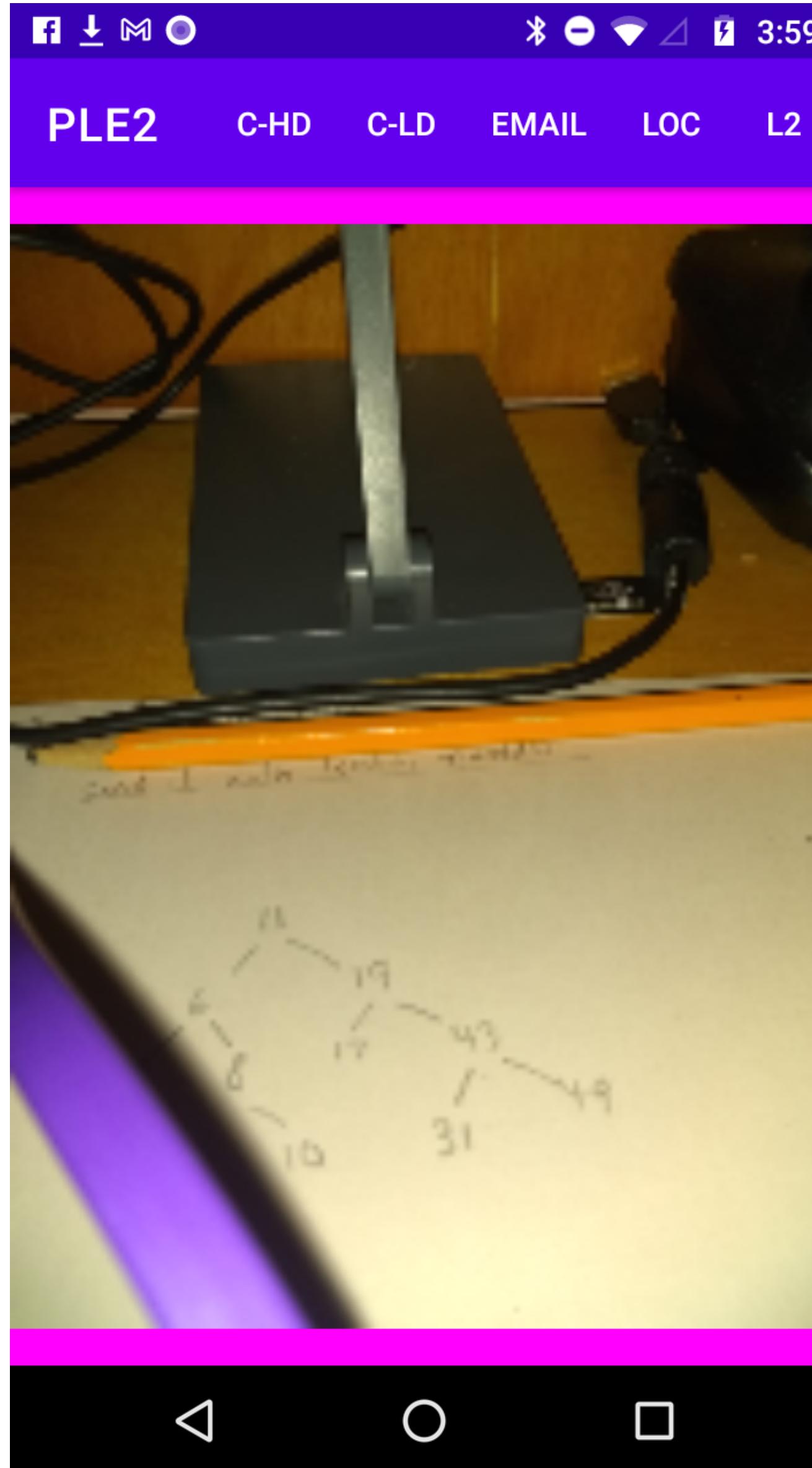
Show the image

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("THIS", "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("THIS", "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

- FileProvider — new(ish) android annoyance.
 - Security model no longer has any shared file space.
 - But to get a hi-res image you need to pass to photo app a place to write the image.
 - So, give it a URI that another activity can write to.
 - (You also 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>
```

```
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("THIS", "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 pointer to bitmap
 - But I wanted to use the image in other places and easiest way to do that is with a static variable

```
// 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
 - Need to do the same FileProvider thing
 - 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

```
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 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" />

    <data
        android:pathPattern=".*\\.txt"
    />
</intent-filter>
```