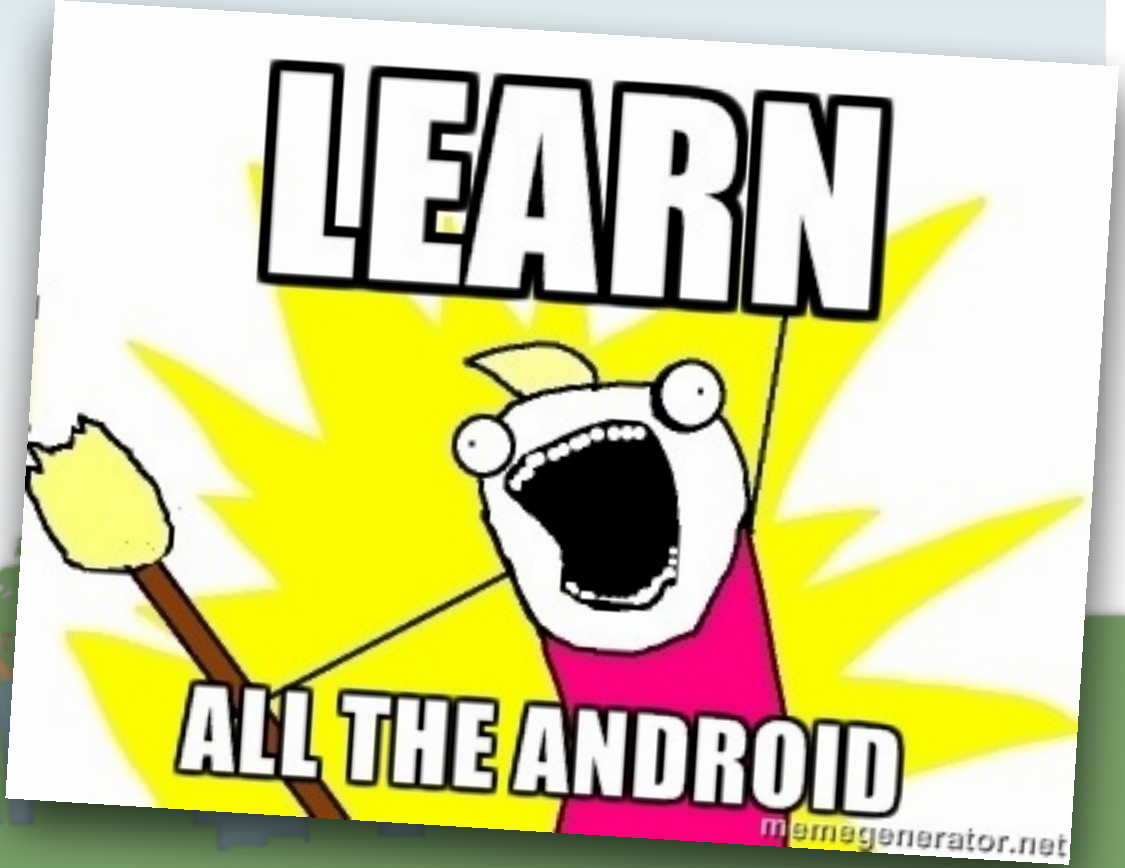
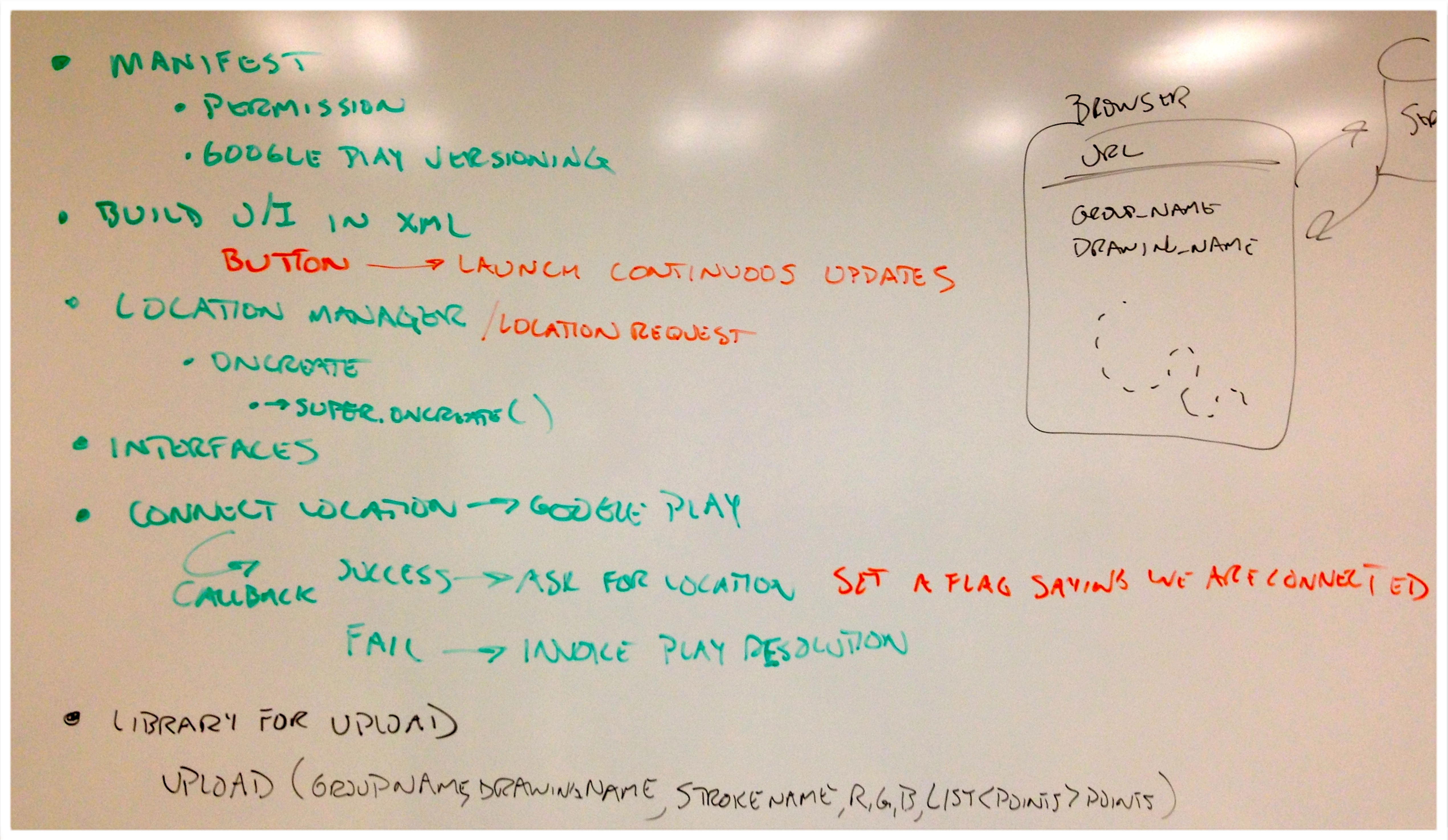
GPS Drawing 3:

Android Continous Location

Mobile and Ubiquitous Games
ICS 163
Donald J. Patterson



GPS Drawing - Changes to making the single location continuous



- MANIFEST
  - PERMISSION
  - GOOGLE PLAY VERSIONING
- BUILD U/I IN XML

  BUTTON → LAUNCH CONTINUOUS UPDATES
- LOCATION MANAGER /LOCATION REQUEST
  - ONCREATE
    - → SUPER.ONCREATE( )
- INTERFACES
- CONNECT LOCATION → GOOGLE PLAY

  CALLBACK

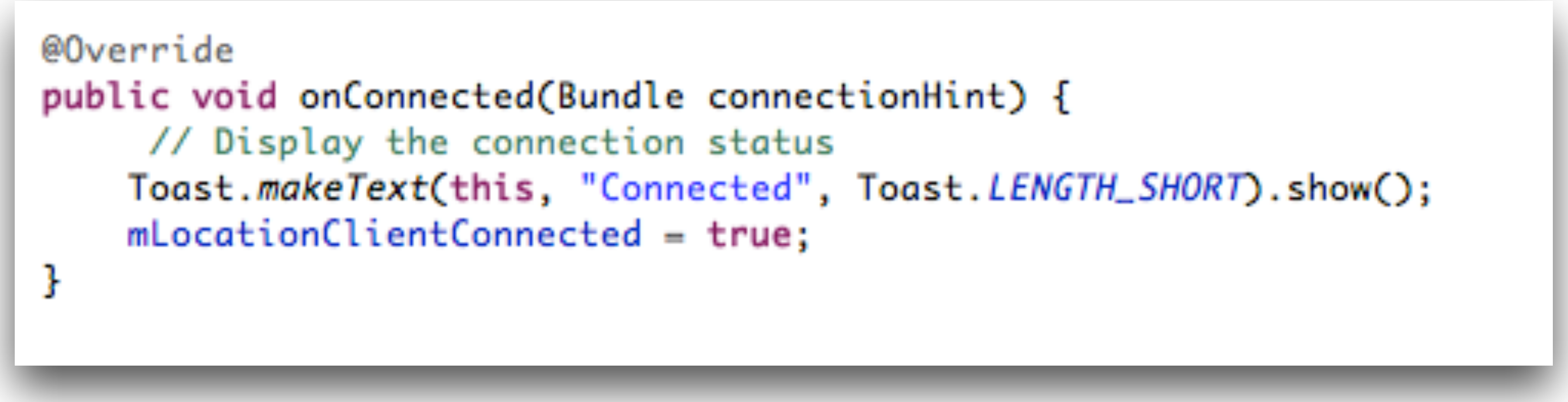
  SUCCESS → ASK FOR LOCATION SET A FLAG SAYING WE ARE CONNECTED

  FAIL → INVOKE PLAY RESOLUTION
- LIBRARY FOR UPLOAD

  UPLOAD (GROUPNAME DRAWINGNAME, STROKENAME, R,G,B, LIST<POINTS> POINTS)

GPS Drawing - Changes to making the single location continuous

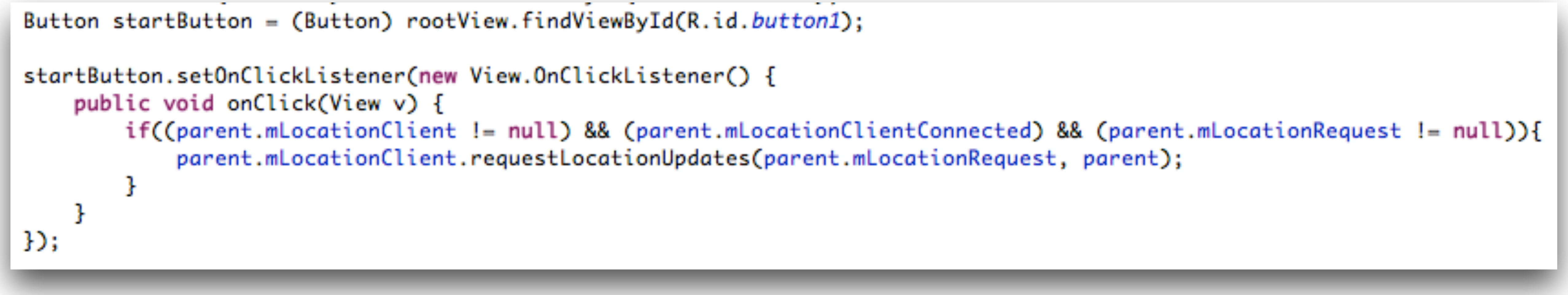
- Instead of getting location on the Play connection call back, you can set a variable. Then a button can be used to initiate the continuous location feed



```
@Override
public void onConnected(Bundle connectionHint) {
     // Display the connection status
    Toast.makeText(this, "Connected", Toast.LENGTH_SHORT).show();
    mLocationClientConnected = true;
}
```

GPS Drawing - Changes to making the single location continuous

- Dynamically assign a function to a button
- When clicked it launches the location call backs

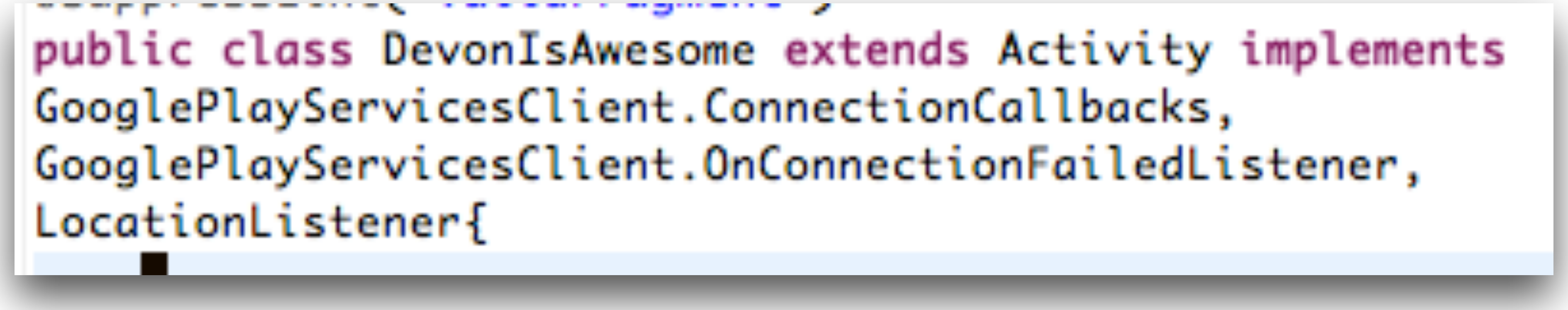


```
Button startButton = (Button) rootView.findViewById(R.id.button1);

startButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if((parent.mLocationClient != null) && (parent.mLocationClientConnected) && (parent.mLocationRequest != null)){
            parent.mLocationClient.requestLocationUpdates(parent.mLocationRequest, parent);
        }
    }
});
```

GPS Drawing - Changes to making the single location continuous

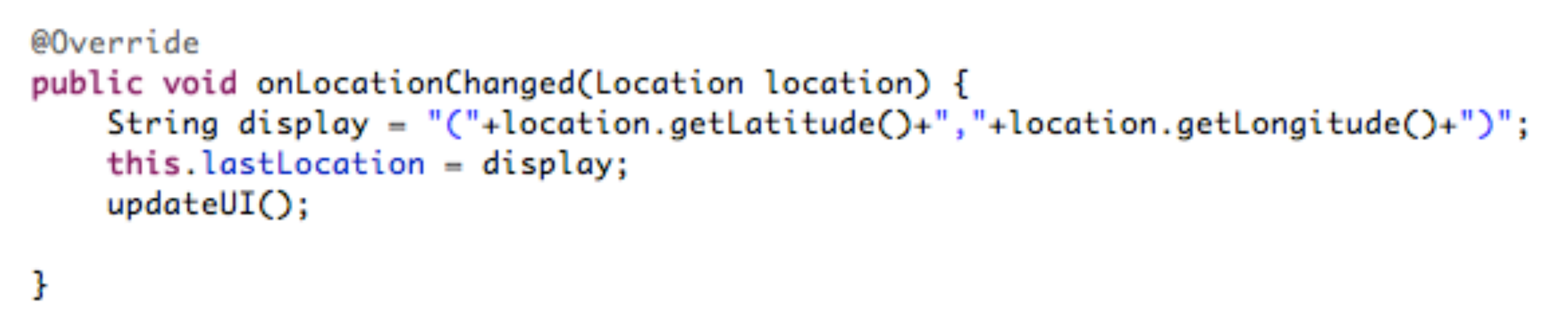
- The location callbacks are defined by the LocationListener interface



```
public class DevonIsAwesome extends Activity implements
GooglePlayServicesClient.ConnectionCallbacks,
GooglePlayServicesClient.OnConnectionFailedListener,
LocationListener{
```

GPS Drawing - Changes to making the single location continuous

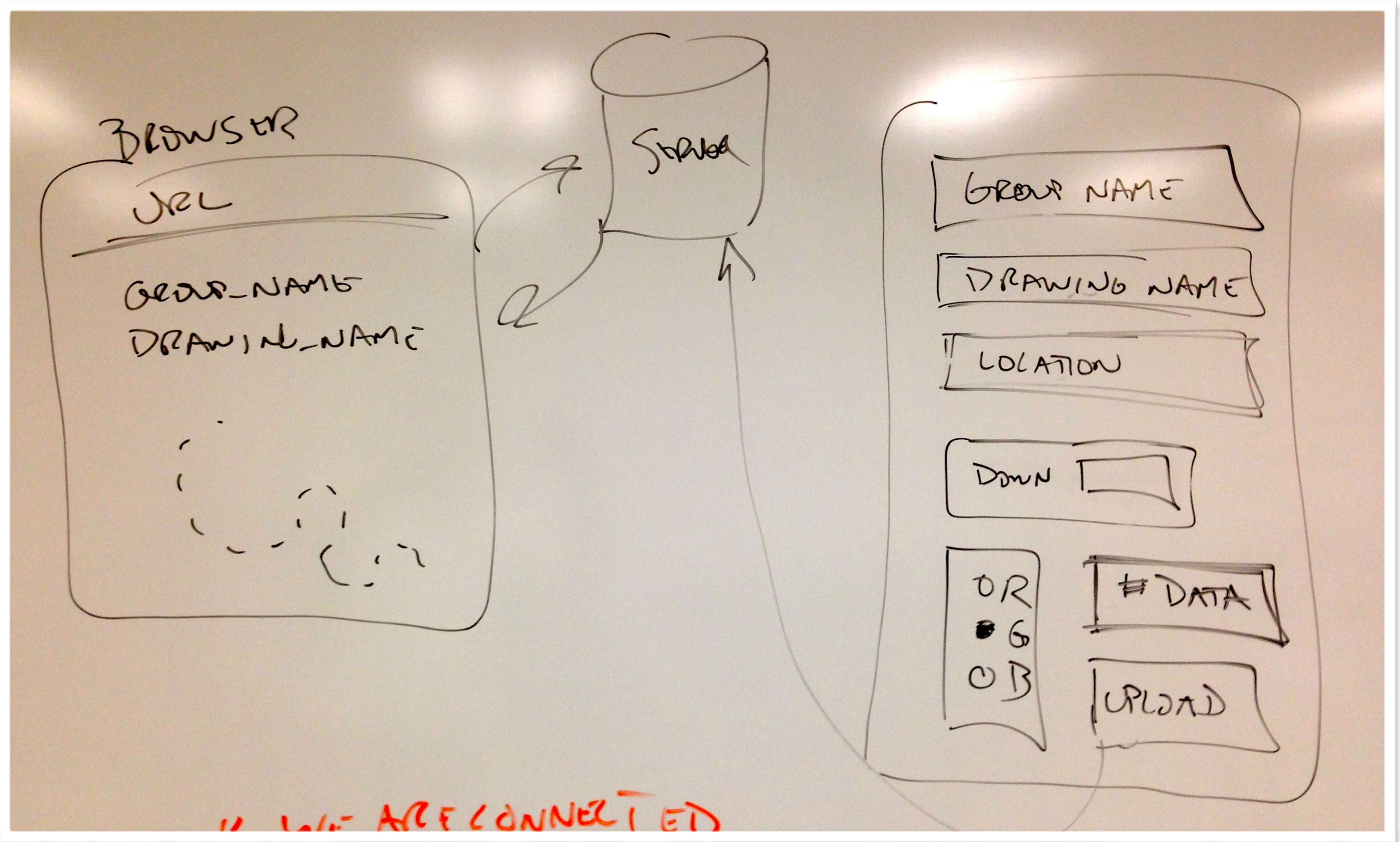
- Which defines the onLocationChanged callback.
- In that function we update our U/I



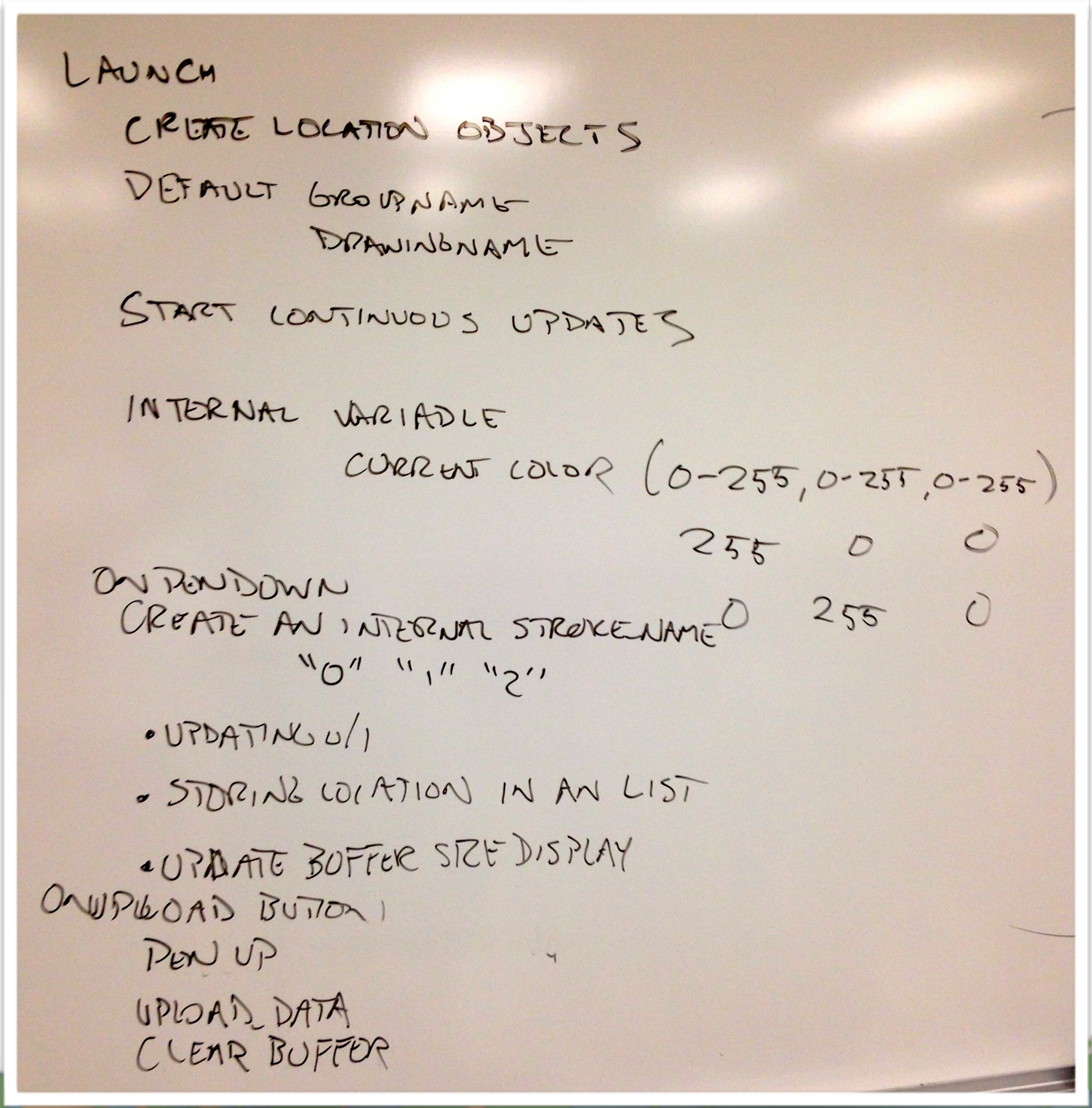
```
@Override
public void onLocationChanged(Location location) {
    String display = "("+location.getLatitude()+","+location.getLongitude()+")";
    this.lastLocation = display;
    updateUI();

}
```

GPS Drawing


BROWSER
URL
GROUP_NAME
DRAWING_NAME
1
2
SERVER
GROUP NAME
DRAWING NAME
LOCATION
DOWN
○ R
● G
○ B
# DATA
UPLOAD
WE ARE CONNECTED

GPS Drawing - The flow for the assignment



```
LAUNCH
    CREATE LOCATION OBJECTS
    DEFAULT GROUPNAME
            DRAWINGNAME

    START CONTINUOUS UPDATES

    INTERNAL VARIABLE
            CURRENT COLOR (0-255, 0-255, 0-255)
                              255     0     0
  ON PENDOWN
    CREATE AN INTERNAL STROKENAME  0     255     0
            "0"  "1"  "2"

      • UPDATING 0/1
      • STORING LOCATION IN AN LIST
      • UPDATE BUFFER SIZE DISPLAY
ON UPLOAD BUTTON
      PEN UP
      UPLOAD DATA
      CLEAR BUFFER
```

GPS Drawing - Changes to making the single location continuous

- In the final assignment you will have to keep track of three additional things (primarily):
  - A color that the user selects
  - Whether the pen is up or down
  - A collection that holds the points that you are observing when the pen is down

- The final bit about how to upload is on it's way

LUCI