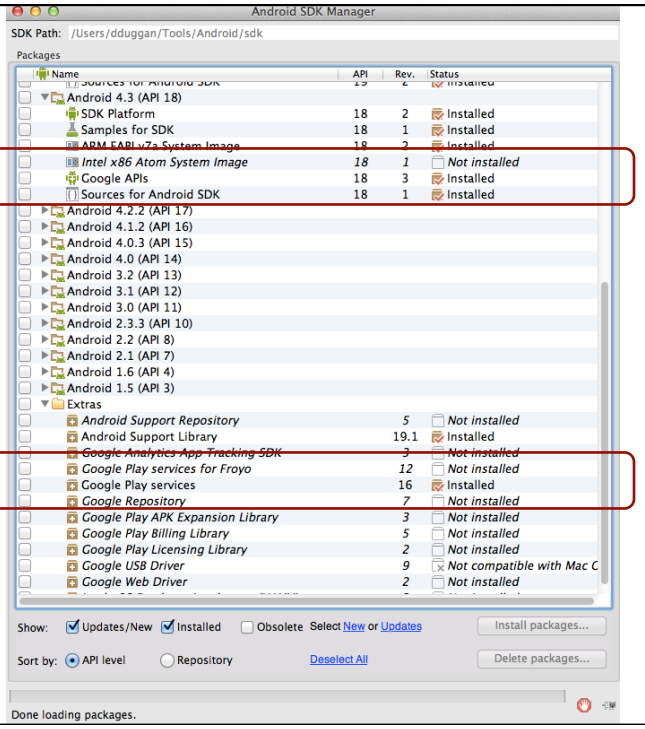
Location APIs

Dominic Duggan
Stevens Institute of Technology

Installing Google Play Services

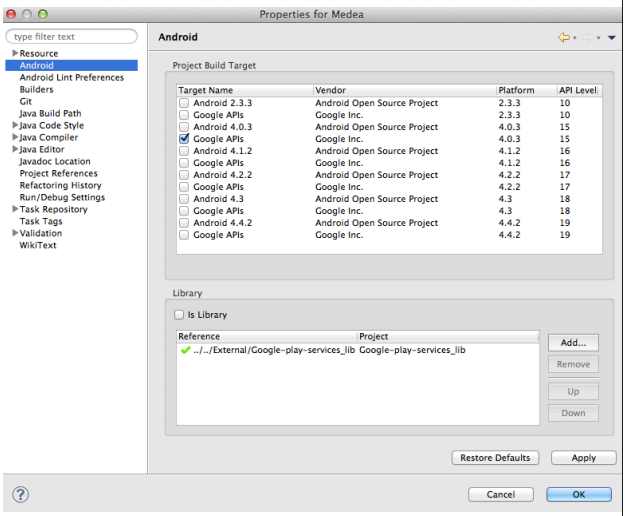
- From Android SDK:
  - Download Google APIs for Android version
  - Create AVD for Google APIs
- From Android SDK:
  - Download Google Play Services (extra)
- Copy Google Play Services Library to Eclipse projects space
  - Import into workspace



3

Installing Google Play Services

- Reference Play Services library from app:
  - Properties | Android | Library



4

Installing Google Play Services

- Add version information to app manifest:

```
<meta-data
   android:name=
      "com.google.android.gms.version"
   android:value=
     "@integer/google_play_services_version" />
```

5

Installing Google Play Services

- Create Proguard exception:

```
-keep class * extends java.util.ListResourceBundle {
    protected Object[][] getContents();
}

-keep public class
com.google.android.gms.common.internal.safeparcel.SafeParcelable {
    public static final *** NULL;
}

-keepnames @com.google.android.gms.common.annotation.KeepName class *
-keepclassmembernames class * {
    @com.google.android.gms.common.annotation.KeepName *;
}

-keepnames class * implements android.os.Parcelable {
    public static final ** CREATOR;
}
```

6

Installing Google Play Services

- Check Play Services APK version

```
public static boolean checkPlayServices(Activity context) {
    int resultCode = GooglePlayServicesUtil
                      .isGooglePlayServicesAvailable(context);
    if (resultCode != ConnectionResult.SUCCESS) {
        if (GooglePlayServicesUtil
             .isUserRecoverableError(resultCode)) {
            GooglePlayServicesUtil.getErrorDialog(resultCode, context,
                                        PLAY_SERVICES_RESOLUTION_REQUEST)
                                  .show();
        } else {
            Log.e(TAG, "This device is not supported.");
            context.finish();
        }
        return false;
    }
    return true;
}
```

7

LOCATION API

8

Permissions

- ACCESS_COARSE_LOCATION
- ACCESS_FINE_LOCATION

```
<uses-permission
   android:name=
     "android.permission.ACCESS_COARSE_LOCATION"/>
```

9

Location Client

```
public class MainActivity extends Activity implements
        GooglePlayServicesClient.ConnectionCallbacks,
        GooglePlayServicesClient.OnConnectionFailedListener {
    ...
    public void onConnected(Bundle dataBundle) { /* Connected to service */ }
    ...
    public void onDisconnected() {  /* Connection dropped due to error */ }
    ...
    public void onConnectionFailed(ConnectionResult connectionResult) {
        /* Location service fails */
        if (connectionResult.hasResolution()) {
           // Start an Activity that tries to resolve the error
           connectionResult.startResolutionForResult(
                       this,
                       CONNECTION_FAILURE_RESOLUTION_REQUEST);
        } else {
        }
    }
}
```

10

Location Client

- Create in onCreate()

```
LocationClient locationClient;
locationClient =
   new LocationClient(this, this, this);
```

- Connect in onStart()

```
locationClient.connect();
```

- Get current location:

```
Location current =
   LocationClient.getLastLocation();
```

- Disconnect in onStop()

```
locationClient.disconnect();
```

11

Location Updates

- Define callback:

```
public class MainActivity extends Activity implements
        GooglePlayServicesClient.ConnectionCallbacks,
        GooglePlayServicesClient.OnConnectionFailedListener,
        LocationListener {
    ...
    @Override
    public void onLocationChanged(Location location) {
        ... Double.toString(location.getLatitude() ...
        ... Double.toString(location.getLongitude());
    }
    ...
}
```

12

Location Updates

- Tune frequency & accuracy
- Update interval

```
LocationRequest.setInterval();
```

- Fastest update interval
  - Avoid data overflow
  - Save power

```
LocationRequest.setFastestInterval();
```

13

Location Updates

- Tune frequency & accuracy
- Update interval

```
LocationRequest.setInterval();
```

- Fastest update interval
  - Avoid data overflow
  - Save power
  - Adjust down if long-running (e.g. network) work

```
LocationRequest.setFastestInterval();
```

14

Location Updates

```
public static final int UPDATE_INTERVAL_IN_SECONDS = 5;
private static final long UPDATE_INTERVAL =
          UPDATE_INTERVAL_IN_SECONDS * 1000;
private static final int FASTEST_INTERVAL_IN_SECONDS = 1;
private static final long FASTEST_INTERVAL =
          FASTEST_INTERVAL_IN_SECONDS * 1000;

LocationRequest locationRequest = LocationRequest.create();
locationRequest
    .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
locationRequest.setInterval(UPDATE_INTERVAL);
locationRequest.setFastestInterval(FASTEST_INTERVAL);
```

15

Requesting Location Updates

- Create location client on onCreate()
- Connect to location service in onStart()
- Request updates in onConnected()

```
LocationClient locationClient;
LocationRequest locationRequest;
boolean userAllowsUpdates;

public void onConnected(Bundle dataBundle) {
   if (userAllowsUpdates) {
      locationClient
       .requestLocationUpdates(locationRequest, this);
   }
}
```

16

Requesting Location Updates

- Create location client on onCreate()
- Connect to location service in onStart()
- Request updates in onConnected()

```
LocationClient locationClient;
LocationRequest locationRequest;
boolean userAllowsUpdates;

public void onStop() {
   if (locationClient.isConnected()) {
      locationClient.removeLocationUpdates(this);
      locationClient.disconnect();
   }
}
```

17

Address Lookup

```
Private class GetAddressTask
      extends AsyncTask<Location, Void, String> {
   protected String doInBackground(Location… Params) {
      Geocoder geocoder =
         new Geocoder(context, Locale.getDefault);
      Location loc = params[0];
      List<Address> addresses = geocoder
         .getFromLocation(loc.getLatitude,
                          loc.getLongitude, 1);
      Address address = addresses.get(0);
      String addressText = ...
         ... address.getAddressLine(i)
         ... address.getLocality()
         ... address.getCountryName()
      return addressText;
   }
}
```

18

GEOFENCING

19

Geofencing

- Combine:
  - Current location
  - Proximity to locations of interest
- Mark locations of interest with latitude, longtitude
- Geofence defined by radius from location
  - Also duration
- Request enter and exit events for geofence

20

Creating a Geofence

- Geofence.builder:
  - Latitude, longitude, radius
  - Expiration time
  - Transition type (“entry” and “exit”)
  - Geofence ID
- Store geofence flattened
  - Database
  - Storage provider

21

Getting Geofence Transitions

- Pending intent e.g. for IntentService:

```
private PendingIntent getTransitionPendingIntent() {
    // Create an explicit Intent
    Intent intent = new Intent(this,
            ReceiveTransitionsIntentService.class);
    /*
     * Return the PendingIntent
     */
    return PendingIntent.getService(
            this,
            0,
            intent,
            PendingIntent.FLAG_UPDATE_CURRENT);
}
```

22

Sending Monitoring Requests

- Interfaces:

  `ConnectionCallbacks`

  – Connection to Location Services

  `OnConnectionFailedListener`

  – If error while attempting to connect

  `OnAddGeofencesResultListener`

  – Once a geofence has been added

23

Sending Monitoring Requests

```
private void onConnected(Bundle dataBundle) {
   ...
   switch (requestType) {
      case ADD :
        // Get the PendingIntent for the request
        transitionPendingIntent =
                        getTransitionPendingIntent();
        // Send a request to add the current geofences
        locationClient.addGeofences(
              currentGeofences, pendingIntent, this);
            ...
        }
    }
```

24

Sending Monitoring Requests

```
public void onAddGeofencesResult(
            int statusCode, String[] geofenceRequestIds) {
   // If adding the geofences was successful
   if (LocationStatusCodes.SUCCESS == statusCode) {
      /*
       * E.g. send out a broadcast intent or update the UI.
       */
   } else {
      // If adding the geofences failed
   }

   // Turn off the in progress flag and disconnect client
   inProgress = false;
   locationClient.disconnect();
}
```

25

Handling Geofence Transitions

```
public class ReceiveTransitionsIntentService extends IntentService
{
   protected void onHandleIntent(Intent intent) {
   if (!LocationClient.hasError(intent)) {
      int transitionType =
         LocationClient.getGeofenceTransition(intent);
      if ( (transitionType == Geofence.GEOFENCE_TRANSITION_ENTER)
        || (transitionType == Geofence.GEOFENCE_TRANSITION_EXIT)) {

         List <Geofence> triggerList =
            LocationClient.getTriggeringGeofences(intent);

         String[] triggerIds = new String[geofenceList.size()];
         // Store the Id of each geofence
         for (int i = 0; i < triggerIds.length; i++) {
             triggerIds[i] = triggerList.get(i).getRequestId();
         // Do something with the list of geofence
       }
    }
```

26

Stopping Geofence Monitoring

- Callback interface:
  - LocationClient.OnRemoveGeofencesResultListener
- Remove all geofences:

```
removeGeofences(PendingIntent,
    LocationClient.OnRemoveGeofencesResultListener)
```

- Callback:

```
onRemoveGeofencesByPendingIntentResult
```

- Remove a collection of geofences:

```
removeGeofences(List<String>
    LocationClient.OnRemoveGeofencesResultListener)
```

- Callback:

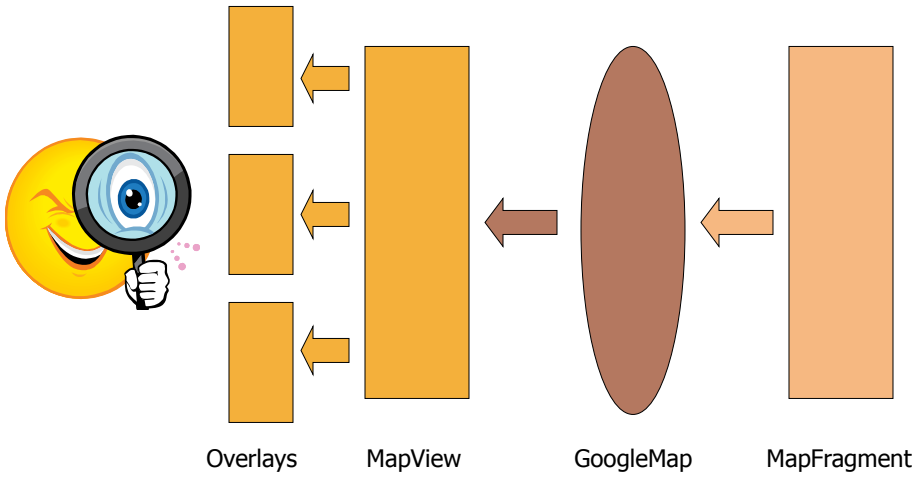
```
onRemoveGeofencesByRequestIdsResult
```

27

MAPS

28

Map-Based Activities



29

Map Fragment

- Create map fragment

```
mapFragment = MapFragment.newInstance();
```

- Add into activity

```
FragmentTransaction fragmentTransaction =
     getFragmentManager().beginTransaction();
fragmentTransaction
    .add(R.id.map_container, mMapFragment);
fragmentTransaction.commit();
```

- Extract map object

```
GoogleMap map = mapFragment.getMap();
```

30

Map Fragment

- Alternatively declare map fragment in layout

```
<fragment
  android:id="@+id/map"
  android:name=
      "com.google.android.gms.maps.MapFragment"
  android:layout_width="match_parent"
  android:layout_height="match_parent" />
```

- Get reference to map fragment resource by resid

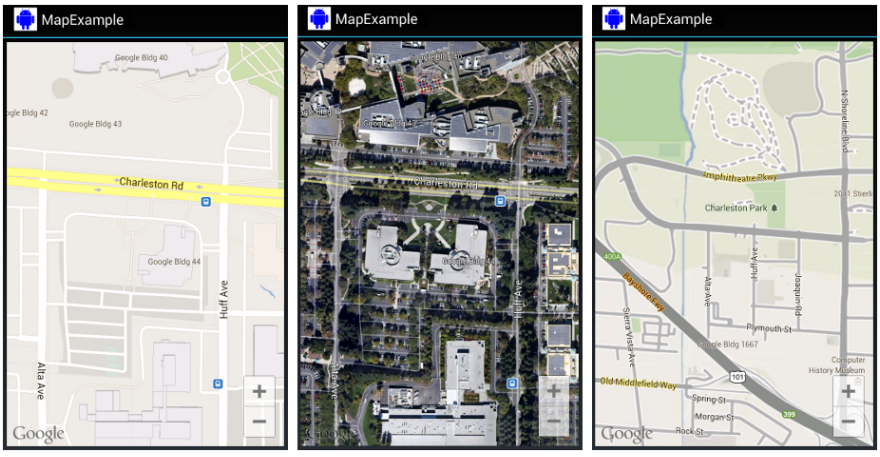
```
mapFragment = (MapFragment)
     getFragmentManager().findFragmentById(R.id.map);
```

- Extract map object

```
GoogleMap map = mapFragment.getMap();
```

31

Types of Maps



32

Types of Maps

- Normal
- Satellite
- Hybrid
- Topographic

```
GoogleMap map;
map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
```

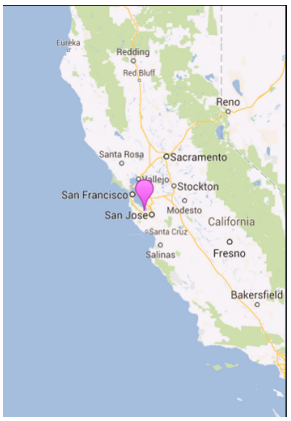
33

Map Attributes

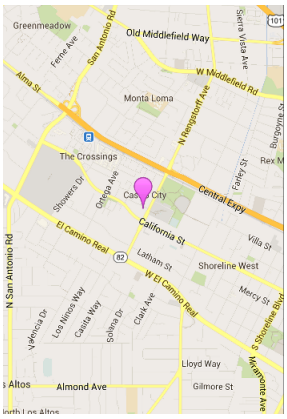
- Map type
- Camera position
  - Location: `cameraTargetLat, cameraTargetLng`
  - Zoom: `cameraZoom`
  - Bearing: `cameraBearing`
  - Tilt: `cameraTilt`
- Enable/disable zoom buttons & compass
- Gestures

34

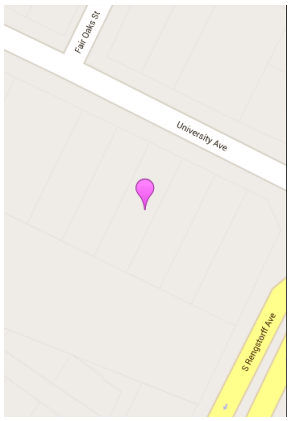
Camera Zoom



Zoom level 6



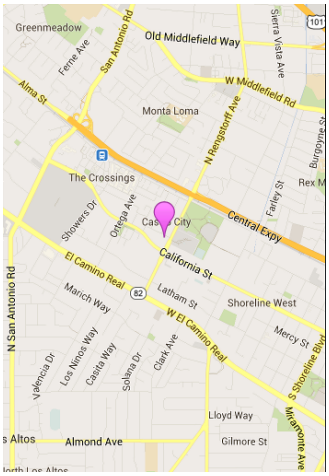
Zoom level 14

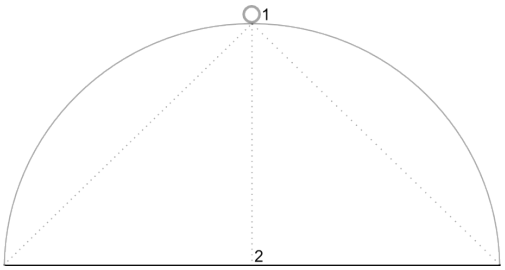


Zoom level 19

35

Camera Tilt

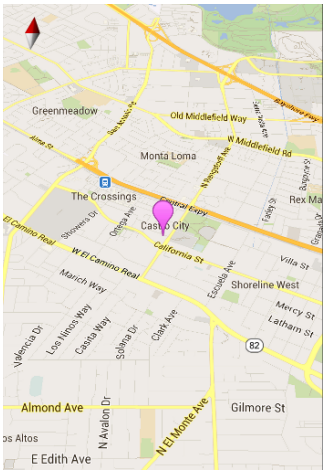


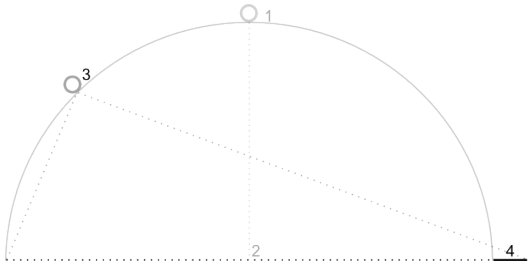


Default (0 degrees viewing angle)

36

Camera Tilt





45 degree (viewing angle)

37

```
<fragment
  xmlns:android=
       "http://schemas.android.com/apk/res/android"
  xmlns:map="http://schemas.android.com/apk/res-auto"
  android:id="@+id/map"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  class="com.google.android.gms.maps.MapFragment"
  map:cameraBearing="112.5"
  map:cameraTargetLat="-33.796923"
  map:cameraTargetLng="150.922433"
  map:cameraTilt="30"
  map:cameraZoom="13"
  map:mapType="normal"
  map:uiCompass="false"
  map:uiRotateGestures="true"
  map:uiScrollGestures="false"
  map:uiTiltGestures="true"
  map:uiZoomControls="false"
  map:uiZoomGestures="true"/>
```

38

Setting Options Programmatically

```
GoogleMapOptions options =
   new GoogleMapOptions();

Options
   .mapType(GoogleMap.MAP_TYPE_SATELLITE)
   .compassEnabled(false)
   .rotateGesturesEnabled(false)
   .tiltGesturesEnabled(false);

MapFragment.newInstance(options)

new MapView(context, options)
```

39

Camera Update

- Changing zoom Level

```
CameraUpdateFactory.zoomIn()
CameraUpdateFactory.zoomOut()

CameraUpdateFactory.zoomTo(float)

CameraUpdateFactory.zoomBy(float)

CameraUpdateFactory.zoomBy(float, Point)
```

40

Camera Update

- Changing camera position & zoom Level

```
CameraUpdateFactory.newLatLng(LatLng)

CameraUpdateFactory
   .newLatLngZoom(LatLng, float)

CameraUpdateFactory
   .newCameraPosition(CameraPosition)
new CameraPosition.Builder()
```

41

Camera Update

- Set boundaries (greatest possible zoom level)

```
CameraUpdateFactory
   .newLatLngBounds(LatLngBounds bounds,
                    int padding)

CameraUpdateFactory
   .newLatLngBounds(LatLngBounds bounds,
                    int width, int height,
                    int padding)
```

42

Camera Update

- Panning
  - +x: move camera right
  - +y: move camera down (!)
  - Direction based on current bearing

```
CameraUpdateFactory.scrollBy(float, float)
```

43

Applying Camera Update

- Animating

```
GoogleMap.moveCamera(cameraUpdate)

GoogleMap.animateCamera(cameraUpdate)

GoogleMap
  .animateCamera(cameraUpdate,
                 callback,
                 duration)
```

44

Example

```
LatLng HOBOKEN = new LatLng(47.4, -74.0);

CameraPosition cameraPosition =
   new CameraPosition.Builder()
     .target(HOBOKEN)
     .zoom(17)
     .bearing(90) // Face east
     .tilt(30)
     .build();

map.animateCamera
   (CameraUpdateFactory
      .newCameraPosition(cameraPosition));
```

45

ACCESSING MAPS API

46

Step 1: Device Certificate

- Application certificate used to sign apps
  - Debug certificate: `debug.keystore`
    - Linux, MacOS: `~/.android/debug.keystore`
    - Windows: `C:\Users\<user>\.android\debug.keystore`
  - Release certificate:

```
keytool -genkey -v -keystore my-keys.keystore
   -alias alias-name -keyalg RSA -keysize 2048
   -validity 10000
```

  - Further information:

```
http://developer.android.com/tools/publishing/app-
signing.html
```

47

Step 1: Device Certificate

- Maps API key based on SHA-1 fingerprint
  - Debug:

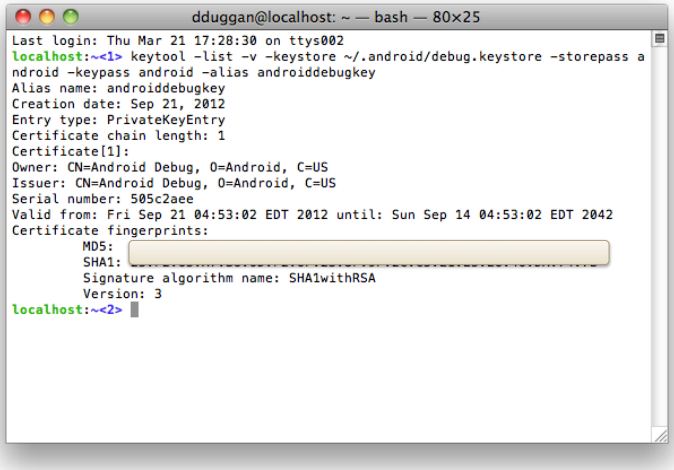
```
keytool -list -v -keystore ~/.android/debug.keystore
-storepass android -keypass android
-alias androiddebugkey
```

  - Release:

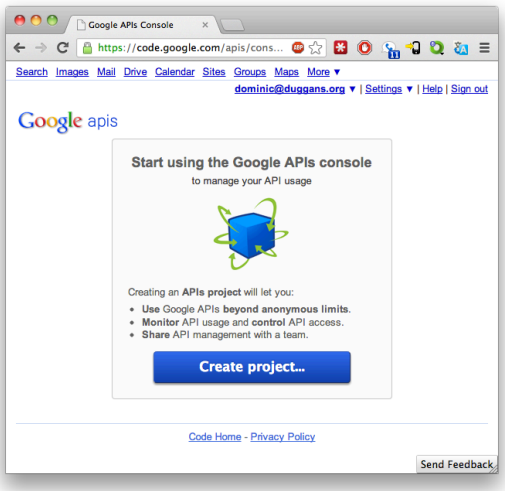
```
keytool -list -v -keystore your_keystore_name
–storepass keystore-password –keypass key-password
-alias your_alias_name
```

48

Step 1: Device Certificate

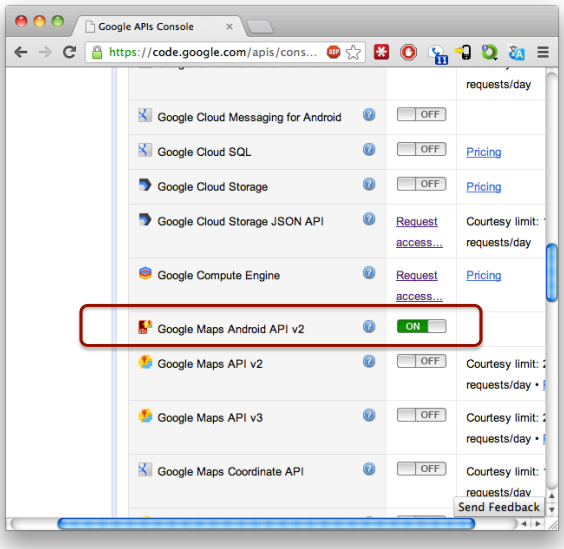


Step 2: Create API Project



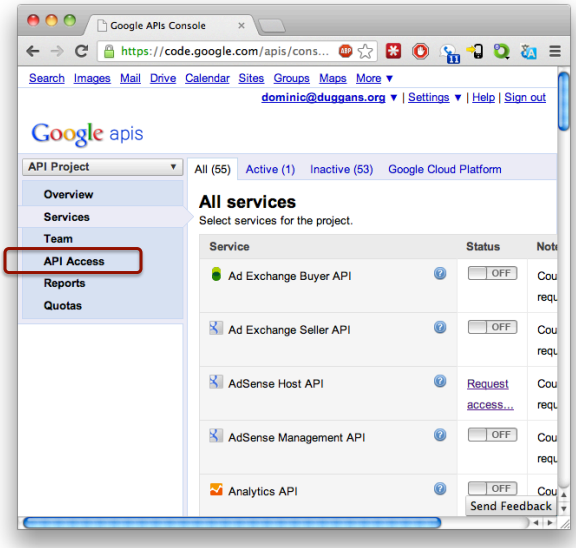
https://code.google.com/apis/console/

Step 2: Create API Project



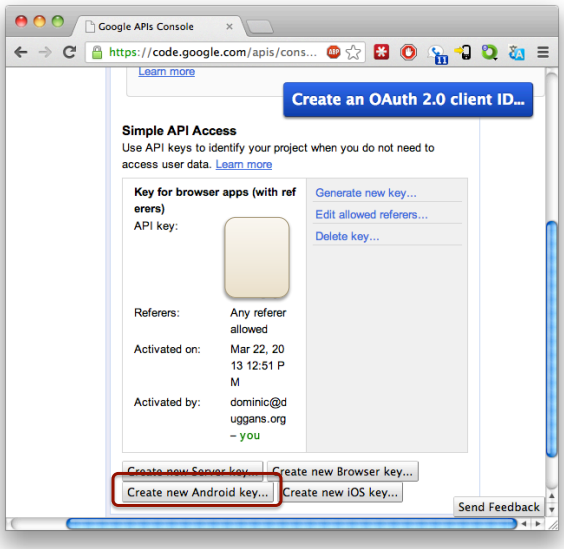
51

Step 3: Obtaining an API Key

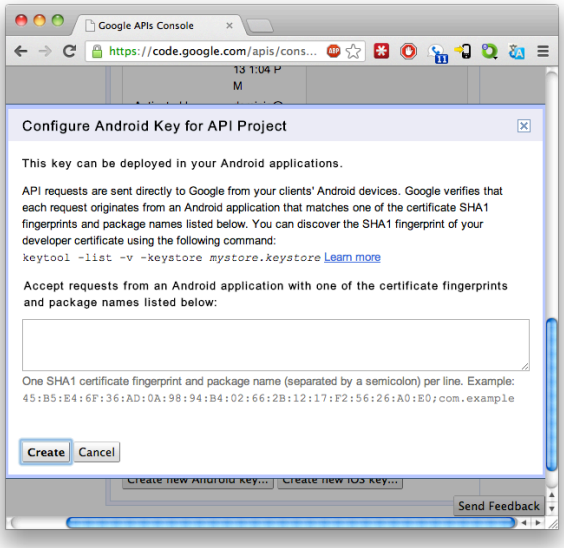


52

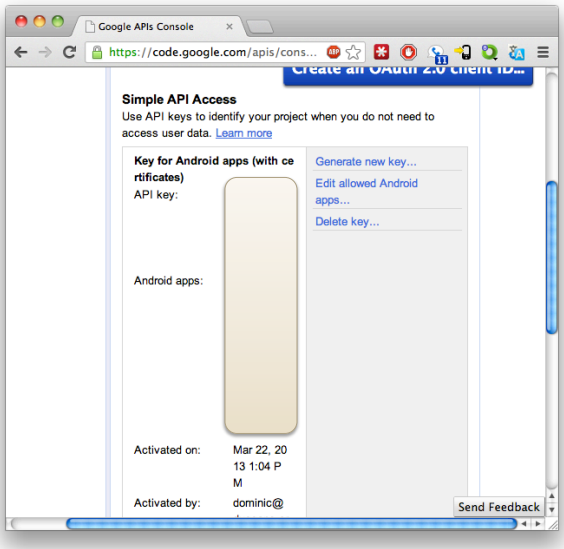
Step 3: Obtaining an API Key



Step 3: Obtaining an API Key



Step 3: Obtaining an API Key



55

Step 4: Application Manifest

- API Key:

```
<application>
   ...
   <meta-data
      android:name=
         "com.google.android.maps.v2.API_KEY"
      android:value="api_key"/>
</application>
```

- Require OpenGL ES Version 2

```
<uses-feature
   android:glEsVersion="0x00020000"
   android:required="true"/>
```

56

Step 4: Application Manifest

- App-specific permission:

```
<permission
  android:name=
    "com.example.mapdemo.permission.MAPS_RECEIVE"
  android:protectionLevel="signature"/>

<uses-permission
  android:name=
    "com.example.mapdemo.permission.MAPS_RECEIVE"/>
```

57

Step 4: Application Manifest

```
<permission
  android:name="com.example.mapdemo.permission.MAPS_RECEIVE"
  android:protectionLevel="signature"/>
<uses-permission android:name="com.example.mapdemo.permission.MAPS_RECEIVE"/>
<!-- Copied from Google Maps Library/AndroidManifest.xml. -->
<uses-sdk
  android:minSdkVersion="8"
  android:targetSdkVersion="16"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<!-- External storage for caching. -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<!-- My Location -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<!-- Maps API needs OpenGL ES 2.0. -->
<uses-feature
  android:glEsVersion="0x00020000"
  android:required="true"/>
```

58

Step 4: Permissions

```
android.permission.INTERNET

com.google.android.providers
   .gsf.permission.READ_GSERVICES

android.permission.ACCESS_NETWORK_STATE

android.permission.WRITE_EXTERNAL_STORAGE

<uses-feature
        android:glEsVersion="0x00020000"
        android:required="true"/>
```

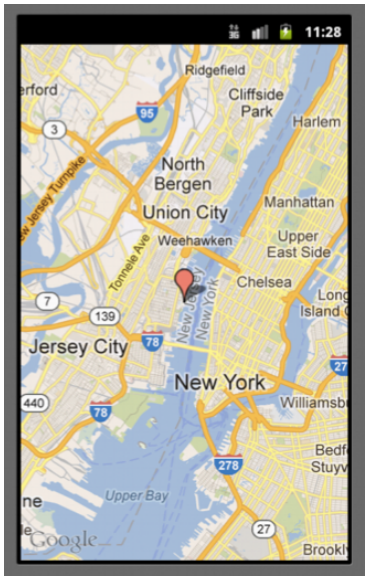
59

INTERACTING WITH MAPS

60

Markers

```
GoogleMap map;

map.addMarker(

   new MarkerOptions()

      .position(
       new LatLng(...))

      .title(
       "My location")

      .icon(...));
```

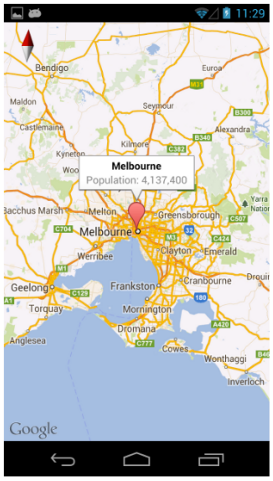


Marker Customization

- Position
- Title
- Snippet
- Draggable
- Visible
- Anchor
- Icon

62

Changing Icon




```
map.addMarker(

   new MarkerOptions()

     .position(new LatLng(-37.81319,
                          144.96298))

     .title(“Melbourne”)

     .snippet("Population: 4,137,400”)

     .icon(BitmapDescriptorFactory
           .defaultMarker
              (BitmapDescriptorFactory.HUE_AZURE)));
```

63

Changing Icon

```
BitmapDescriptorFactory
   .fromAsset(String assetName)
   .fromBitmap (Bitmap image)
   .fromFile (String path)
   .fromResource (int resourceId)
```

```
map.addMarker(

   new MarkerOptions()

     .position(new LatLng(-37.81319,
                          144.96298))

     .title(“Melbourne”)

     .snippet("Population: 4,137,400”)

     .icon(BitmapDescriptorFactory
           .fromResource(R.drawable.arrow)));
```

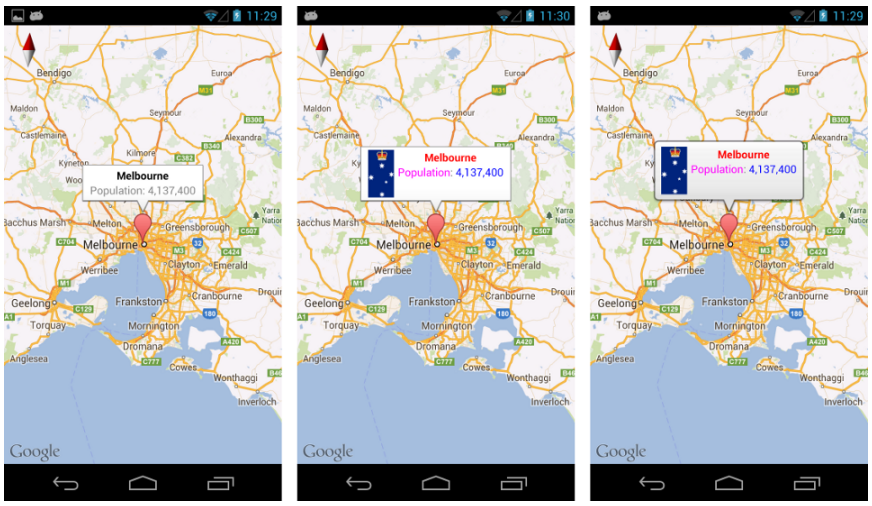
64

Changing Info Window

```
InfoWindowAdapter adapter =
   new InfoWindowAdapter() {
      public InfoWIndow getInfoWindow() {
         /* Default info window */
         /* factory method */
      }
      public InfoWIndow getInfoContents() {
         /* Called if other method returns null */
      }
   };

GoogleMap map;
map.setInfoWindowAdapter(adapter);
```

65

Changing Info Window



66

Events

- Marker click events
  `onMarkerClick(Marker)`
- Marker drag events
  `onMarkerDragListener`
  `onMarkerDragStart(Marker)`
  `onMarkerDrag(Marker)`
  `onMarkerDragEnd(Marker)`
  `Marker.getPosition()`
- Info window click events
  `onInfoWindowClick(Marker)`

67

Shapes

- Polyline
- Polygon
- Circle

```
// Add a rectangle
PolygonOptions rectOptions = new PolygonOptions()
              .add(new LatLng(37.35, -122.0),
                   new LatLng(37.45, -122.0),
                   new LatLng(37.45, -122.2),
                   new LatLng(37.35, -122.2),
                   new LatLng(37.35, -122.0));

Polygon polygon = map.addPolygon(rectOptions);
```

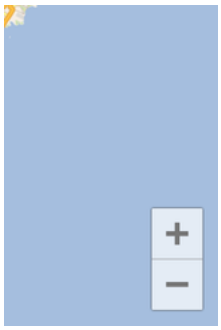
68

Donuts

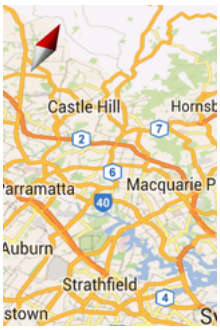
```
Polygon(
   new PolygonOptions()
        .add(new LatLng(0, 0),
             new LatLng(0, 5),
             new LatLng(3, 5),
             new LatLng(3, 0),
             new LatLng(0, 0))
        .addHole(new LatLng(1, 1),
                 new LatLng(1, 2),
                 new LatLng(2, 2),
                 new LatLng(2, 1),
                 new LatLng(1, 1))
        .fillColor(Color.BLUE));
```

69

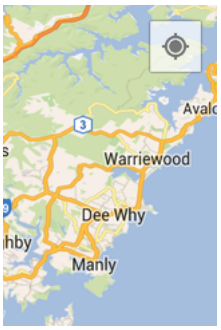
Map Controls



Zoom



Compass



Location

70

Map Gestures

- Zoom
- Scroll (pan)
- Tilt
- Rotate

71

Map Events

- Click
- Long click
- Camera change

72

LOCATION API

73

Location Providers

```
LocationManager locationManager =
  (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);
```

- Common Location Providers:
  - LocationManager.GPS_PROVIDER
  - LocationManager.NETWORK_PROVIDER

```
Location location =
  locationManager.getLastKnownLocation
    (LocationManager.GPS_PROVIDER);

boolean enabledOnly = true;
List<String> providers =
   locationManager.getProviders(enabledOnly);
```

74

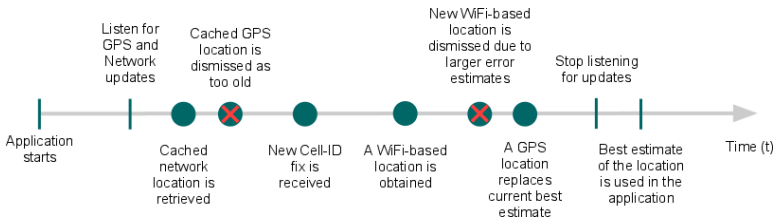
Criteria

```
Criteria criteria = new Criteria();
criteria.setAccuracy(10);
criteria.setAltitudeRequired(false);
criteria.setBearingRequired(false);
criteria.setCostAllowed(true);
criteria.setPowerRequirement(Criteria.NO_REQUIREMENT);
criteria.setSpeedRequired(false);

String bestProvider =
      locationManager.getBestProvider(criteria, true);

List<String> matchingProviders =
      locationManager.getProviders(criteria, false);
```

75

Criteria

- Return matching location provider with greatest accuracy
- If none match, relax criteria in this order:
  - Power use
  - Accuracy
  - Ability to return bearing, speed, altitude



76

Maintaining Current Best Estimate

```
long timeDelta =
   newLocation.getTime() - currentBestLocation.getTime();

boolean isSignificantlyNewer = timeDelta > TWO_MINUTES;
boolean isSignificantlyOlder = timeDelta < -TWO_MINUTES;
boolean isNewer = timeDelta > 0;

int accuracyDelta = (int)
   (newLocation.getAccuracy() –
         currentBestLocation.getAccuracy());

boolean isLessAccurate = accuracyDelta > 0;
boolean isMoreAccurate = accuracyDelta < 0;
boolean isSignificantlyLessAccurate =
   accuracyDelta > 200;
```

77

Maintaining Current Best Estimate

```
if (isSignificantlyNewer)
    currentBestLocation = newLocation;

else if (isSignificantlyOlder) {
    /* Do nothing */ ;

else if (isMoreAccurate)
    currentBestLocation = newLocation;

else if (isNewer && !isLessAccurate) {
    currentBestLocation = newLocation;

else if (isNewer &&
         !isSignificantlyLessAccurate &&
         isFromSameProvider) {
    currentBestLocation = newLocation;
```

78

Getting Location

```
locationManager = (LocationManager)
      getSystemService(Context.LOCATION_SERVICE);

Criteria criteria = ...

String provider =
      locationManager.getBestProvider(criteria, true);

Location location =
      locationManager.getLastKnownLocation(provider);

... location.getLatitude() ... location.getLongitude() ...
```

79

Getting Location

- Permissions tag

```
<uses-permission android:name=
    “android.permission.ACCESS_FINE_LOCATION”/>
<uses-permission android:name=
   “android.permission.ACCESS_COARSE_LOCATION”/>
```

80

Notification of Location Changes

```
LocationListener myLocationListener = new LocationListener() {

   public void onLocationChanged(Location location) {
        … location.getLatitude() … location.getLongitude() …
   }

   public void onProviderDisabled(String provider){
      ...
   }

   public void onProviderEnabled(String provider){
      ...
   }

   public void onStatusChanged(String provider, int status,
                               Bundle extras){
      ...
   };
```


Notification of Location Changes

```
locationManager
  .requestLocationUpdates(
      provider,
      2000,
      10,
      locationListener);

locationManager
   .removeUpdates(myLocationListener);
```

Time interval in milliseconds

Range of movement in meters

Proximity Alert (1)

```
private void setProximityAlert() {
   LocationManager locationManager = (LocationManager)
      getSystemService(Context.LOCATION_SERVICE);

   double lat = 73.147536;
   double lng = 0.510638;
   float radius = 100f; //meters
   long expiration = -1; //do not expire

   Intent intent = new Intent (TREASURE_PROXIMITY_ALERT);
   PendingIntent proximityIntent =
      PendingIntent.getBroadcast
       (getApplicationContext(), -1, intent, 0);
   locationManager.addProximityAlert
       (lat, lng, radius, expiration, proximityIntent);

   IntentFilter filter =
        new IntentFilter(TREASURE_PROXIMITY_ALERT);
   registerReceiver(new ProximityIntentReceiver(), filter);
 }
```


Proximity Alert (2)

```
PendingIntent proximityIntent =
   PendingIntent
     .getBroadcast (getApplicationContext(), -1, intent, 0);

locationManager.addProximityAlert (lat, lng, radius, expiration,
                                   proximityIntent);

public class ProximityIntentReceiver extends BroadcastReceiver {
     public void onReceive (Context context, Intent intent) {
        String key = LocationManager.KEY_PROXIMITY_ENTERING;

        Boolean entering = intent.getBooleanExtra(key, false);
        Toast.makeText(MyActivity.this, "Treasure: " + entering,
                       Toast.LENGTH_LONG).show();
        // ... perform proximity alert actions ...
     }
}
```

Reverse Geocoding

```
location =
   locationManager.getLastKnownLocation
         LocationManager.GPS_PROVIDER);
double latitude = location.getLatitude();
double longitude = location.getLongitude();

List<Address> addresses = null;

Geocoder geocoder =
   new Geocoder(getApplicationContext(), Locale.getDefault());

try {
   addresses = geocoder.getFromLocation(latitude, longitude, 10);
} catch (IOException e) {}

Toast.makeText(this, addresses.get(0).getCountryName(),
               Toast.LENGTH_LONG).show();
```

85

Forward Geocoding

```
String streetAddress =
   "160 Riverside Drive, New York, New York";

List<Address> locations = null;

Geocoder fwdGeocoder = new Geocoder(this, Locale.US);
try {
   locations =
     fwdGeocoder.getFromLocationName(streetAddress, 10);
} catch (IOException e) {}

String latLngString =
   addresses.get(0).getLatitude() + ", " +
   addresses.get(0).getLongitude();

Toast.makeText(this, latLngString , Toast.LENGTH_LONG)
   .show();
```

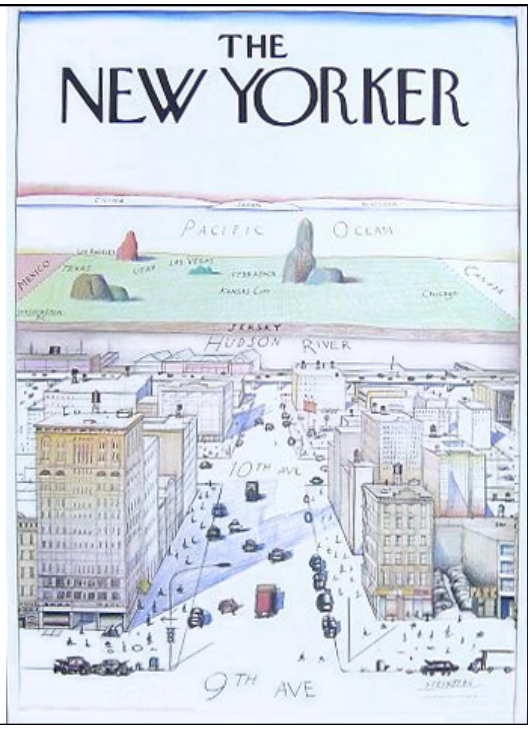
86

CONTEXT-AWARENESS

87

People don't speak GPS

- Different people have different views of the world.



Places -- big and small

- People refer to location as places
  - countries, cities, towns, streets, buildings
  - rooms, spaces within buildings
  - relation to other places,
    - e.g. across from Starbucks
- GPS is too precise and may require accurate map or building plan
  - Jim might be at 42.3325N, -71.11861E but is he in the shower at the moment?

Context Awareness

- Systems "aware of," and respond to, their context (situation, environment, emotion, …)

- Physical context
  - location
  - orientation
  - date and time
  - temperature
  - humidity
  - device
  - capabilities
- Logical context
  - interests
  - work / leisure
  - activity
  - user preferences

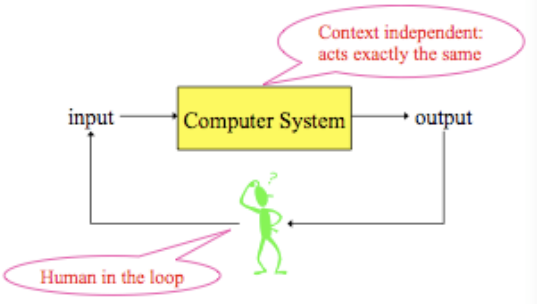
90

Issues: Context Awareness

- Needed for an environment minimally intrusive
  - Recognize user state and surroundings
  - Make decisions proactively, modify behavior accordingly
- Issues:
  - Obtaining information needed to function
  - How to represent context internally? How to combine it with system and application state? Where to store?
  - How often to update and consult context information?
  - What services does infrastructure have to provide?
  - How to track location? sense surroundings?

91

Traditional View of Computer Systems



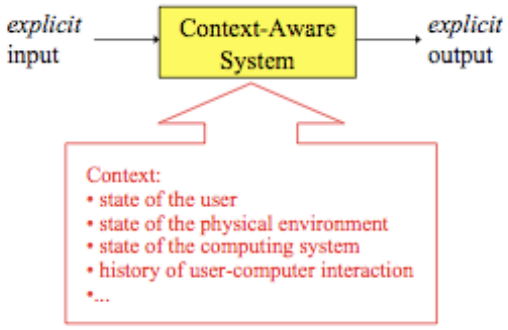
92

How to make systems smarter?

- "Intelligence" with AI
- "Good" human-computer interface
- Doing the right thing
  ⇒ requiring sensitivity to "context"
  - sense/acquire
  - learn/deduce
  - act/adapt

93

Context as *Implicit* Input/Output



94

From Abstraction to Context Sensitivity

- Traditional black box view comes from the desire for *abstraction*
- This is based on several assumptions:
  - Explicit input/output: slow, intrusive, requiring user attention
  - Sequential input-output loop
- Move away from the black box model and into **context-sensitivity**
  - human out-of-the-loop (as much as possible)
  - reduce explicit interaction (as much as possible)

95

Context-Aware Computing

- Let computer systems sense automatically, remember history, and adapt to changing situations
  - Reduced explicit interaction, more responsive
- Need to draw a boundary around the system under consideration
  - To define "explicit" and "implicit"

96

CONTEXT-AWARE COMPUTING

97

Context

- Many definitions:
  - Where a person is, who one is with, and what resources are nearby
  - Any information that can be used to characterize the situation of an entity
- Example: Museum guide
  - Other people are a part of the context but the weather is not

98

Definition of Context (1)

- Schilit:
  - **Computing context :** network connection, communication costs, nearby resources (displays, printers, …)
  - **User context:** user's profile, location, people nearby, current activity
  - **Physical context:** lighting, noise level, traffic conditions
- Time is also important and natural context
  - **Time context**, or
  - History

99

Definition of Context (2)

- Schmidt et al.: "knowledge about user's and IT device's state, including surroundings, situation, and to a less extent, location"
- Dey: "any information that can be used to characterize the situation of an entity"
  - Entity: person, place, object that is considered relevant

100

Definition of Context (3)

- Kotz: "the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user"
- **Active context:** influences behavior of an application
  - Presentation of information and services to a user
  - Selection of services or information for a user
  - Automatic execution of a service for a user
- **Passive context:** relevant to the application, but not critical
  - Tagging of context to information for later retrieval

101

Examples of Context

- Identity, activity, feeling
- Spatial: location, orientation, speed
- Temporal: date, time of day, season
- Environmental: temperature, light, noise
- Social: people nearby, activity, calendar
- Resources: nearby, availability
- Physiological: blood pressure, heart rate, tone of voice

102

Context Awareness

- A system is context-aware if it uses context to provide **relevant** information and/or services to the user, where relevance depends on the user's **task**
- Reasons to be context-aware
  - Context-specific services and applications
    - Position of persons and things
    - Selection and Filtering of information
  - Overcome limitations
    - Restricted user interfaces
    - Limited electrical power
    - Limited communication range
    - Limited mobility
  - **Situational awareness**

Context Scenarios

- Telecommunications
  - Depending on the number of people in a room, a mobile phone decides to ring normally or to vibrate only
  - Roaming the device (from wired phone to mobile)
- Presentations
- Human-Computer Interaction
  - Using the best input/output devices within reach (e.g. Minority Report)

Context-Aware Computing (1)

- Pascoe: taxonomy of context-aware features
  - contextual sensing
  - context adaptation
  - contextual resource discovery
  - contextual augmentation (associating digital data with user's context)

105

Context-Aware Computing (2)

- Dey: context-aware features
  - presentation of information/services to a user according to current context
  - automatic execution of a service when in a certain context
  - tagging context to information for later retrieval

106

Context-Aware Computing

- Kotz:
  - Active context awareness - An application automatically adapts to discovered context
  - Passive context awareness - An application presents the new or updated context

107

Context-Aware Pervasive System

- Loke:
  - Applications — **Acting Subsystem**
  - Storage/Management
  - Preprocessing/reasoning — **Thinking Subsystem**
  - Raw data retrieval
  - Sensors — **Sensing Subsystem**

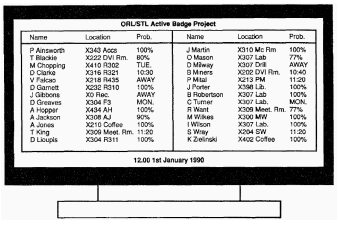
108

CONTEXT-AWARE APPLICATIONS

109

Call Forwarding

- Location of employees is shown to receptionist
  - Receptionist forwards calls to the nearest phone
  - Eventually, receptionist was removed from the loop
- Is this a good idea?
  - Important meeting? (or taking a nap!)
  - Private call is forwarded to a non private room?
  - "Context-aware communication"
- What is the context?



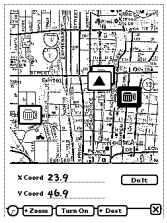
Shopping Assistant

- Scenario: you are at home depot trying to figure out what you need to finish your basement
- Shopping assistant can
  - Tell you what parts you need
  - Where to find them relative to your location in the store
  - What is on sale
  - Do comparative pricing
  - Use your previous profile information to customize shopping and delivery
- What context? Active/passive?

111

CyberGuide

- Characteristic
  - From Georgia Tech
  - Provide information services to a tourist
    - Direction
    - Background information
    - Travel diary (automatically recorded)
  - Indoor and Outdoor Version
    - Indoor
      - Using IR (infrared)
    - Outdoor
      - Using GPS



Figure：Outdoor Cyberguide Screenshot, etc.♪

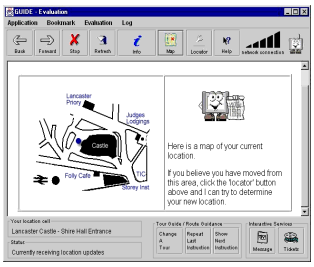
112

Augmented Reality

- User's view of the real world is augmented with additional information
- Scenario 1: you wear sun-glasses with a display and headphones and walk in downtown Rome
  - Sistine Chapel
- Scenario 2: special ops squad infiltrating an enemy complex wearing the same gizmos
  - Night vision
- Context?

GUIDE

- Developed in Lancaster University
  - For Lancaster City visitors



Screenshot of GUIDE♪

- Using WaveLAN as communication infrastructure
  - A tourist comes to a region(cell)
  - Receives information of the region.

Adaptive GSM phone/PDA

- PDA: notepad application changes its characteristics depending on user activity
  - Large font when walking, small font when stationary
  - Change the intensity level depending on the lighting conditions
- Phone: decide ring volume or vibration depending on situation
  - In hand, in a suitcase, on a table, in a classroom/ conference?

115

Conference Assistant

- Assistant checks out conference schedule and user's interest to highlight which presentations should be attended
- Automatically present speaker names, title of paper/other relevant info
- May record slides, audio, questions, take notes and tag them with the appropriate place in the talk for later retrieval

116

Location Aware Information Delivery

- Generate reminders to users as they enter locations
  - Voice synthesis used to generate audio reminders
  - "Sorry, dear, I forgot" is going to become obsolete?
- CyberMinder (GaTech) uses a complex context including nearby people, weather conditions
- But what is the basic context needed here?

117

Observations

- Human-Computer interaction can be improved with context info
  - Intelligent applications? AI?
- Focus on simple context(s): e.g., location, identity, time
- Why aren't other contexts used more heavily?
  - Not that useful?
  - Difficult to sense/model?
  - Need more imagination?
  - A natural consequence of these being mobile applications?

118

StartleCam

- From MIT Media Lab
- Composed with
  - Wearable video camera
  - Computer
  - sensing system
- Save Image when the wearer is interested
  - By sensing skin conductivity signal

Figure：StartleCam System♪

119

CAMP-UP

- Phone manager
  - Functionality database
- Space manager
  - Policy database
  - Context interpreter
- Device exposes functionality to the space
- Space sends context information to device
  - Location
  - Activity
- Space will require certain functionality of the device

120