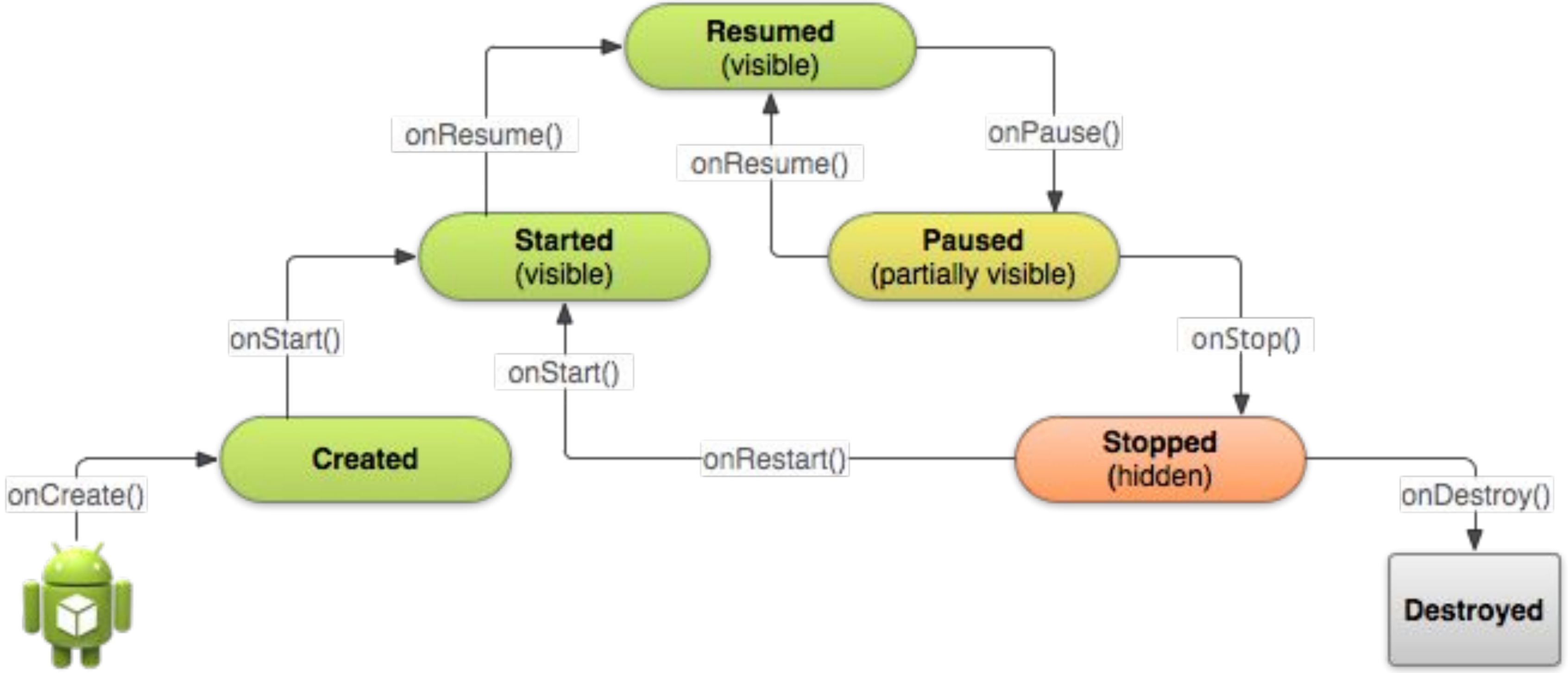
5COSC005W MOBILE APPLICATION DEVELOPMENT
Lecture 4: More on the Activity Lifecycle and Implicit Intents

Dr Dimitris C. Dracopoulos

Module Web page:

http://users.wmin.ac.uk/∼dracopd/DOCUM/courses/5cosc005w/5cosc005w.html

Activity states and callbacks graph



Saving instance state

Implement `onSaveInstanceState()` in your Activity

- Called by Android runtime when there is a possibility the Activity may be destroyed
- Saves data only for this instance of the Activity during current session
- `onSaveInstanceState` is not called when user explicitly closes the activity (e.g. presses the Back button) or when `finish()` is called. Use `onPause()` or `onStop()` instead

onSaveInstanceState(Bundle outState)

```
@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    // Add information for saving HelloToast counter
    // to the to the outState bundle
    outState.putString("count",
                String.valueOf(mShowCount.getText()));
}
```

Restoring instance state

Two ways to retrieve the saved Bundle

- in `onCreate(Bundle mySavedState)`
  Preferred, to ensure that your user interface, including any saved state, is back up and running as quickly as possible
- Implement callback (called after onStart())
  [`onRestoreInstanceState(Bundle mySavedState)`]

Restoring in onCreate()

```
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);
   mShowCount = findViewById(R.id.show_count);

   if (savedInstanceState != null) {
       String count = savedInstanceState.getString("count");
       if (mShowCount != null)
           mShowCount.setText(count);
   }
}
```

Implicit Intents

What can an Intent do?

An `Intent` can be used to:

- start an Activity
- start a Service
- deliver a Broadcast

Services and Broadcasts will be covered in other lessons

Explicit vs. implicit Intent

Explicit Intent — Starts an Activity of a specific class

Implicit Intent — Asks system to find an Activity class with a registered handler that can handle this request

What you do with an implicit Intent

- Start an Activity in another app by describing an action you intend to perform, such as "share an article", "view a map", or "take a picture"
- Specify an action and optionally provide data with which to perform the action
- Don't specify the target Activity class, just the intended action

What system does with implicit Intent

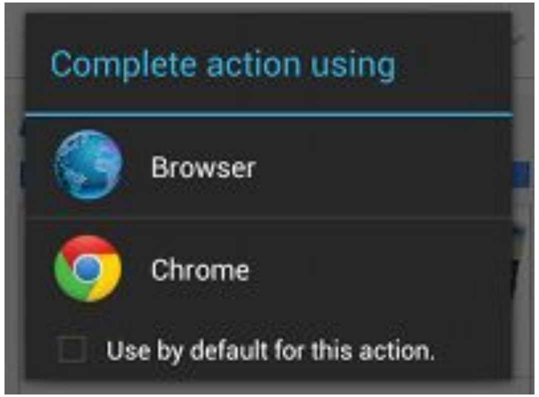
- Android runtime matches the implicit intent request with registered intent handlers
- If there are multiple matches, an App Chooser will open to let the user decide

How does implicit Intent work?

1. The Android Runtime keeps a list of registered Apps
2. Apps have to register via AndroidManifest.xml
3. Runtime receives the request and looks for matches
4. Android runtime uses Intent filters for matching
5. If more than one match, shows a list of possible matches and lets the user choose one
6. Android runtime starts the requested activity

App Chooser

When the Android runtime finds multiple registered activities that can handle an implicit Intent, it displays an App Chooser to allow the user to select the handler



Sending an implicit Intent

Sending an implicit Intent

1. Create an Intent for an action

```
Intent intent = new Intent(Intent.ACTION_CALL_BUTTON);
```

   User has pressed Call button — start Activity that can make a call (no data is passed in or returned)

2. Start the Activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Avoid exceptions and crashes

Before starting an implicit Activity, use the package manager to check that there is a package with an Activity that matches the given criteria.

```
Intent myIntent = new Intent(Intent.ACTION_CALL_BUTTON);

if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Sending an implicit Intent with data URI

1. Create an Intent for action

```
Intent intent = new Intent(Intent.ACTION_DIAL);
```

2. Provide data as a URI

```
intent.setData(Uri.parse("tel:8005551234"));
```

3. Start the Activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Providing the data as URI

Create an URI from a string using `Uri.parse(String uri)`

- `Uri.parse("tel:8005551234")`
- `Uri.parse("geo:0,0?q=brooklyn%20bridge%2C%20brooklyn%2C%20ny")`
- `Uri.parse("http://www.android.com");`

Uri documentation

Implicit Intent examples

Show a web page

```
Uri uri = Uri.parse("http://www.google.com");
Intent it = new Intent(Intent.ACTION_VIEW,uri);
startActivity(it);
```

Dial a phone number

```
Uri uri = Uri.parse("tel:8005551234");
Intent it = new Intent(Intent.ACTION_DIAL, uri);
startActivity(it);
```

Sending an implicit Intent with extras

1. Create an Intent for an action

```
Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);
```

2. Put extras

```
String query = edittext.getText().toString();
intent.putExtra(SearchManager.QUERY, query));
```

3. Start the Activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Category

Additional information about the kind of component to handle the intent.

- `CATEGORY_OPENABLE`
  Only allow URIs of files that are openable
- `CATEGORY_BROWSABLE`
  Only an Activity that can start a web browser to display data referenced by the URI

Sending an implicit Intent with type and category

1. Create an Intent for an action

```
Intent intent = new Intent(Intent.ACTION_CREATE_DOCUMENT);
```

2. Set mime type and category for additional information

```
intent.setType("application/pdf"); // set MIME type
intent.addCategory(Intent.CATEGORY_OPENABLE);
```

continued on next slide...

Sending an implicit Intent with type and category

3. Start the Activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
  startActivityForResult(intent,ACTIVITY_REQUEST_CREATE_FILE);
}
```

4. Process returned content URI in `onActivityResult()`

Common actions for an implicit Intent

Common actions include:

- ACTION_SET_ALARM
- ACTION_IMAGE_CAPTURE
- ACTION_CREATE_DOCUMENT
- ACTION_SENDTO
- and many more

Apps that handle common actions

Common actions are usually handled by installed apps (both system apps and other apps), such as:

- Alarm Clock, Calendar, Camera, Contacts
- Email, File Storage, Maps, Music/Video
- Notes, Phone, Search, Settings
- Text Messaging and Web Browsing

→ List of common actions for an implicit intent

→ List of all available actions

Receiving an Implicit Intent

Register your app to receive an Intent

- Declare one or more Intent filters for the Activity in AndroidManifest.xml
- Filter announces ability of Activity to accept an implicit Intent
- Filter puts conditions on the Intent that the Activity accepts

Intent filter in AndroidManifest.xml

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```

Intent filters: action and category

- `action` — Match one or more action constants
  - `android.intent.action.VIEW` — matches any Intent with `ACTION_VIEW`
  - `android.intent.action.SEND` — matches any Intent with `ACTION_SEND`

- `category` — additional information (list of categories)
  - `android.intent.category.BROWSABLE`—can be started by web browser
  - `android.intent.category.LAUNCHER`—Show activity as launcher icon

Intent filters: data

- `data` — Filter on data URIs, MIME type
    - `android:scheme="https"`—require URIs to be https protocol
    - `android:host="developer.android.com"`—only accept an Intent from specified hosts
    - `android:mimeType="text/plain"`—limit the acceptable types of documents

An Activity can have multiple filters

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    ...
  </intent-filter>
  <intent-filter>
    <action android:name="android.intent.action.SEND_MULTIPLE"/>
    ...
  </intent-filter>
</activity>
```

An Activity can have several filters

A filter can have multiple actions & data

```
<intent-filter>
  <action android:name="android.intent.action.SEND"/>
  <action android:name="android.intent.action.SEND_MULTIPLE"/>
  <category android:name="android.intent.category.DEFAULT"/>
  <data android:mimeType="image/*"/>
  <data android:mimeType="video/*"/>
</intent-filter>
```