

COMP08076

Programming Native App Interaction

- Module coordinator: John Nixon
 - john.nixon@uws.ac.uk, room E259, x3617
- Lecture 4
 - Multi-activity app
 - Assessment 1

Event Driven Programming

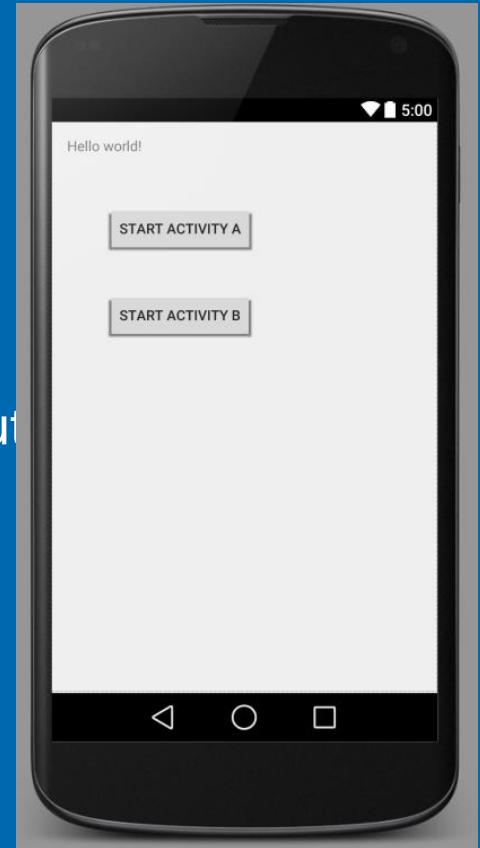
- Programming paradigm where program flow determined by events
 - User interaction events
 - E.g. button presses, taps
 - Events from other threads or processes
 - E.g. phone activity
- Listeners – have set these up for components

Switch statement

- How to determine which button has been pressed?
- The same onClick() method needs to deal with both buttons

- Button id's toActA and toActB

- ```
switch (view.getId()) {
 case R.id.toActA:
 Toast.makeText(this, "Button A been clicked.", Toast.LENGTH_LONG).show()
 break;
 case R.id.toActB:
 Toast.makeText(this, "Button B been clicked.", Toast.LENGTH_LONG).show()
 break;
}
```



# Multi-Activity Apps

- Individual app is likely to have several activities
- Other activities will usually be running on the device
- Android goal is to have activities work together seamlessly.
- Activities may come and go as dictated by memory requirements
- An activity needs to be able to
  - save its application-instance state.
  - get its former state back when it restarts
- Saving instance state is handled by `onSaveInstanceState()`.
  - activity can pour whatever data it needs (e.g., the number showing on the
- That instance state is provided again in two places:
  - in `onCreate()` and in `onRestoreInstanceState()`.
- **`protected void onCreate(Bundle savedInstanceState) {`**
- **`super.onCreate(savedInstanceState);`**
- **`setContentView(R.layout.activity_main);`**
- **`}.`**

# New Activities

- Created in same project
  - From file menu
- Same options as when creating new project
  - Choose empty activity for now
- Has java Activity class file
  - Same package name
- Has XML layout file
- Default structure same as original activity
- How to start new activity?

# Intents

- What is an *intent*?
- An *intent* is an action with its associated data payload
- Or a message to communicate an action
- Use intents to start other activities in your app
- Use intents to start activities in external applications
  - E.g. camera, phone
- Explicit and Implicit

# Explicit versus Implicit

- **Explicit intents** specify the component to start by name (the full class name).
  - use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start.
- **Implicit intents** do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it.
  - if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map...

# Explicit Intent

- Used to start a specific activity either within your app or in an app with known class names
- Intent intent;
- 
- `intent = new Intent(this,ActivityA.class);`
- `startActivity(intent);`
- ActivityA must exist

# Passing Data to a New Activity

- Sometimes useful to send data to a new activity
- `putExtras`
- `Intent intent;`  
`intent = new Intent(this, FruitChoice.class);`
- `intent.putExtra("nameTran",sName);`  
`intent.putExtra("chosenFruit", chosenFruit);`
- `startActivity(intent);`

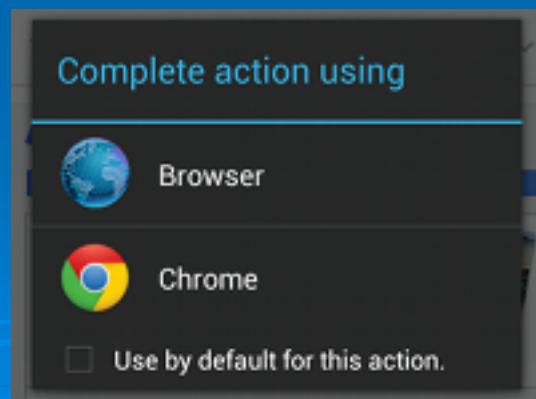
# In the New Activity

- 
- Intent intent = getIntent();
- String uir = intent.getStringExtra("nameTran");  
String uir2 = intent.getStringExtra("chosenFruit");

# Implicit Intent

- Used to start an app by specifying an action
- Uri number = Uri.parse("tel:0141\*\*\*\*\*");
- Intent callIntent;
- callIntent = new Intent(Intent.ACTION\_DIAL, number);
- startActivityForResult(callIntent);
- Specific actions defined within Android
- Caution – on a phone this will work!

- When you call `startActivity()` or `startActivityForResult()` and pass it an implicit intent, the system resolves the intent to an app that can handle the intent and starts its corresponding Activity. If there's more than one app that can handle the intent, the system presents the user with a dialog to pick which app to use



- Uri number = Uri.parse("tel:0141\*\*\*\*\*");  
Intent callIntent = new Intent(Intent.ACTION\_DIAL, number);
  - startActivityForResult(callIntent);
- 
- Uri location = Uri.parse("geo:54.422219,4.08364?z=14");
  - Intent mapIntent = new Intent(Intent.ACTION\_VIEW, location);
  - startActivityForResult(mapIntent);
- 
- Uri webpage = Uri.parse("http://news.bbc.co.uk");
  - Intent webIntent = new Intent(Intent.ACTION\_VIEW, webpage);
  - startActivityForResult(webIntent);

# Common Intents

- <http://developer.android.com/guide/components/intents-common.html>
- Camera
  - ACTION\_IMAGE\_CAPTURE or ACTION\_VIDEO\_CAPTURE
- Contacts
  - ACTION\_PICK
- Email
  - ACTION\_SENDTO (for no attachment) or  
ACTION\_SEND (for one attachment) or  
ACTION\_SEND\_MULTIPLE (for multiple attachments)
- Text
  - ditto

# Checking appropriate app exists

- PackageManager packageManager = getPackageManager();
- List<ResolveInfo> activities = packageManager.queryIntentActivities(mapIntent, 0);
- **boolean** isIntentSafe = activities.size() > 0;
- Toast.makeText(this, String.valueOf(isIntentSafe) ,Toast.LENGTH\_LONG).show();
- 
- 
- If (isIntentSafe){
- startActivity(mapIntent);
- }

# Getting a Result from an Activity

- `startActivityForResult()` instead of `startActivity()`
- ```
Intent pickContactIntent = new Intent(Intent.ACTION_PICK,  
Uri.parse("content://contacts"));
```
- ```
pickContactIntent.setType(Phone.CONTENT_TYPE); // Show user only contacts
with phone numbers
```
- `startActivityForResult(pickContactIntent, PICK_CONTACT_REQUEST);`

```
> @Override
> protected void onActivityResult(int requestCode, int resultCode, Intent data) {
> super.onActivityResult(requestCode, resultCode, data);

>
> switch (requestCode) {
> case (PICK_CONTACT_REQUEST): {
> if (resultCode == RESULT_OK) {
> Uri contactData = data.getData();
> Cursor c = getContentResolver().query(contactData, null, null, null,
> null);
> c.moveToFirst();
> String name = c.getString(c.
 getColumnIndexOrThrow(ContactsContract.Contacts.DISPLAY_NAME_PRIMARY));
> c.close();
> TextView tvRes = (TextView) findViewById(R.id.infoShow);
> tvRes.setText(name);
> }
> break;
> }
> default:
> break;
> }
> }
>
```

# Assessment 1

- [http://moodle.uws.ac.uk/pluginfile.php/301244/course/section/89726/PNAI%20assignment%201%202014\\_15%20%281%29.doc](http://moodle.uws.ac.uk/pluginfile.php/301244/course/section/89726/PNAI%20assignment%201%202014_15%20%281%29.doc)
- Minimum to show competency
- Three screens – so three activities
  - Splash screen
  - Data selection screen
  - Data confirmation screen

- Splash screen
- Navigation to other two screens
  - i.e. can start up each of the other two activities
  - Can use buttons
  - Start of today's lab

- Data selection screen
  - Where most of the work is done
- Use of layouts
- Allows users to enter some data according to the brief
- Does some calculation based on the input data – perhaps when a button is pressed
- Outputs the result of the calculation to the screen

**Book Online Now**

Park Name  
Tantallon Camping and Caravan Pa ▾

Type of Accommodation  
Tourer / Motorhome

Nº of Nights      Arrival Date  
1      20/02/2015 

Adults      Under 16  
1      0

Under 5      Pets  
0      0

At least 1 member of the party must be over 21

**CHECK AVAILABILITY**

- Data confirmation screen
  - Shows some manipulated information from input
- Can be navigated to from the splash screen

# Comments

- // B00\*\*\*\*\*
- // Declaration: I declare that the work submitted is my own unless otherwise stated.
  
- // use comments to explain code
- // to say if something does not work properly
  
- // to acknowledge where ideas have come from
- // such as tutorials
  
- /\* block comment over several lines
- \*/

# Enhancements

- Using shared preferences to store data in data selection screen and display on data confirmation screen
- Dealing with all the data in the brief
- Well designed splash screen using, for example, ImageView
- Improved navigation between screens
  - Back navigation, navigation triggered by different components...
- Increased range of components
  - Text fields, radio buttons, spinners, checkbox....
- Dynamic calculation of cost
  - Triggered by appropriate component events
- Good use of comments
- ....

# Final submission

- 7zip project folder and submit to moodle
- You will be contacted if there is a problem with your submitted project
- If you need to tell us something for marking add a word document into the zipped submission.