CET 325

lifechanging University of Sunderland

The Top Layer: Layouts and UI Controls

Recap

- Android Architecture
- Android Components
- Event Handling

UI Controls

- Button
- Text View
- Edit Text
- Check Boxes
- Radio Buttons
- Radio Group
- Spinner
- Date Picker
- Toast
- Alert Dialog
- … many more!

UI Controls

- Theory by Example
  - CheckBoxes: Task List Application
  - Radio Buttons: Tip Calculator
  - Spinner: Tip Calculator

RadioButton

Class: RadioButton

Package: android.widget

Extends: android.widget.CompoundButton

Description: A two state widget similar to the CheckBox. However, once it is checked the user cannot uncheck it.

Example Methods:

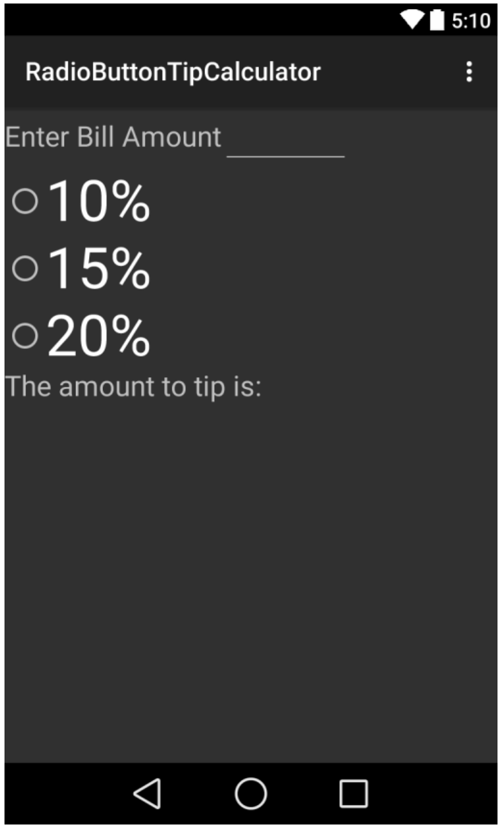
void setOnCheckChangedListener(CompoundButton.OnCheck-ChangedListener occl)

toggle() Forces the state to change state

Boolean isChecked() returns true or false.

lifechanging University of Sunderland

Demo: Tip Calculator



Enter Bill Amount

- ○ 10%
- ○ 15%
- ○ 20%

The amount to tip is:

UI – Radio Group

```
<RadioGroup
    android:id="@+id/tip_choices">

    <RadioButton
       android:id = "@+id/ten"
         android:text = "10%"
         android:textSize = "20pt"/>

    <RadioButton
        android:id = "@+id/twenty"
        android:text = "20%"
        android:textSize = "20pt"/>
</RadioGroup>
```

ALT+Enter –
  Remove hard coded strings

Application Logic

- Declare class variables to represent your components.

```
EditText editTextBillAmount = null;
TextView textViewTipAmount = null;
RadioButton radioButtonTipTen = null;
RadioButton radioButtonTipFifteen = null;
RadioButton radioButtonTipTwenty = null;
RadioGroup rg = null;
DecimalFormat df = new DecimalFormat("£####.00");
```

Application Logic

- Get handle to components in onCreate()

```
editTextBillAmount = (EditText) findViewById(R.id.bill_amount);
textViewTipAmount = (TextView) findViewById(R.id.tip_amount);
radioButtonTipTen = (RadioButton)findViewById(R.id.ten);
radioButtonTipFifteen = (RadioButton)findViewById(R.id.fifteen);
radioButtonTipTwenty = (RadioButton)findViewById(R.id.twenty);
rg = (RadioGroup) findViewById(R.id.tip_choices);
rg.setOnCheckedChangeListener(this);
```

Implement OnCheckedChangeListener and add logic

- This is instead of OnClickListener for buttons

```
public void onCheckedChanged(RadioGroup rg, int i){
    if(i==radioButtonTipTen.getId())
        textViewTipAmount.setText(df.format(Double.parseDouble
                (editTextBillAmount.getText().toString())*.10));

    else if(i==radioButtonTipFifteen.getId())
        textViewTipAmount.setText(df.format(Double.parseDouble
                (editTextBillAmount.getText().toString())*.15));

    else if(i==radioButtonTipTwenty.getId())
        textViewTipAmount.setText(df.format(Double.parseDouble
                (editTextBillAmount.getText().toString())*.20));
}
```

ersity of
lerland

Spinner

Class: Spinner

Package: android.widget

Extends: android.widget.AbsSpinner

Description: A view that displays one child at a time and lets the user pick among them.

Methods:

setOnItemSelectedListener(AdapterView.OnItemSelected, Listener listener)   Assigns a listener to the Spinner. Inherited from AdapterView

setAdapter(SpinnerAdapter adapter)   Associates an adapter to the Spinner. The adapter is the source of the items in the Spinner.

Demo: Tip Calculator

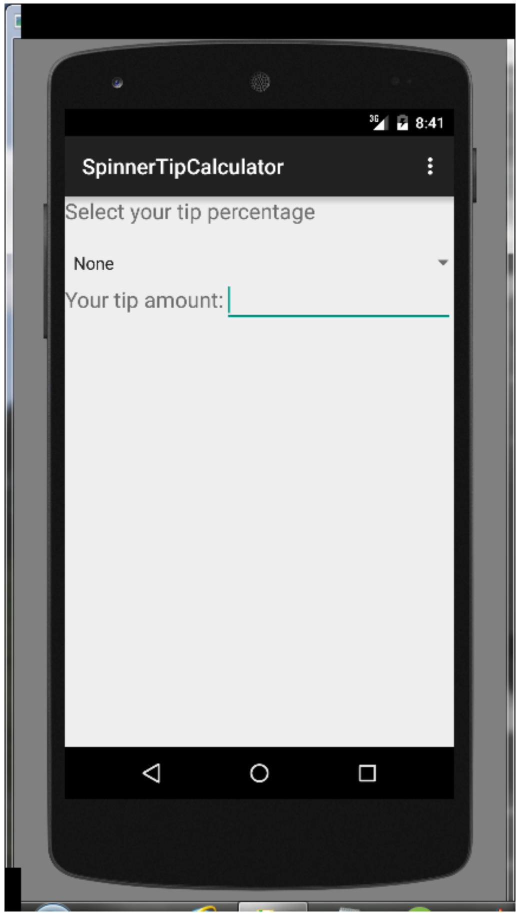
1. Update Strings resource file so that we have an array of possible tip percentages.

```
<string-array name="tip_options">
    <item>Ten Percent</item>
    <item>Fifteen Percent</item>
    <item>Twenty Percent</item>
</string-array>

<string name = "tip_prompt">Select your tip percentage</string>
```

lifechanging University of Sunderland

Demo: Tip Calculator

2. Build your UI



```
<Spinner
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spinner"
    android:entries="@array/tip_options"/>
```

lifechanging University of Sunderland

Demo: Tip Calculator

- Add activity logic

```
Spinner tipSpinner = null;
TextView textViewTipAmount = null;
EditText editTextBillAmount = null;
DecimalFormat df = new DecimalFormat("£####.00");
```

Demo: Tip Calculator

- Add activity logic

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    tipSpinner = (Spinner)findViewById(R.id.spinner);
    editTextBillAmount = (EditText)findViewById(R.id.bill_amount);
    textViewTipAmount = (TextView)findViewById(R.id.tip_amount);

    tipSpinner.setOnItemSelectedListener(this);
}
```

lifechanging University of Sunderland

Demo: Tip Calculator

- Add activity logic
- Listener: AdapterView.OnItemSelectedListener

```
public void onItemSelected(AdapterView<?> parent,
                              View view, int pos, long id){
    //Application logic goes here
    if(id==1){
        textViewTipAmount.setText(df.format(Double.parseDouble
               (editTextBillAmount.getText().toString())*.10));
    }
    else // implement equivalent logic for other options
    else{
        textViewTipAmount.setText("0.00");
    }
}

public void onNothingSelected(AdapterView<?> parent){

}
```

Loading Spinner Content from a List

- Use an ArrayAdapter

```
ArrayList<String> options = new ArrayList<String>();
// fill options
ArrayAdapter<String> dataAdapter;
Spinner spinnerOptions;
spinnerOptions = (Spinner)findViewById(R.id.spinnerOptions);
dataAdapter = new ArrayAdapter<String>
     (this, android.R.layout.simple_spinner_item, options);
```

Can you see a trend?

- Create UI
  - Make IDs
- Declare class level variables in activity
- Instantiate those variables in onCreate()
- Implement the relevant listener and activity logic.
  - Or state the onClick event as an XML attribute
  - Or create an Inner Class

Toast

- Allows runtime user notification.
- Don’t want to waste screen space on something temporary.
- Appears as a burst of text that quickly fades away without user intervention

Toast API

Class: Toast

Package: android.widget

Extends: java.lang.Object

Description: Produces a quick, floating message to the user. Toast never receives focus. Typically used as a confirmation to the user.

Example Methods

Toast MakeText(Context c, CharSequence s, int duration): Makes a standard Toast that just contains a text view. Duration can be LENGTH_LONG or LENGTH_SHORT

void show(): Renders the toast on screen.

void setText(CharSequence t) Updates the Toast Text

Full API: http://developer.android.com/reference/android/widget/Toast.html

Basic Toast Operation

- Instantiate a Toast object with the makeText() method.
- Then display it using show()

```
Context context = getApplicationContext();
CharSequence text = "Hello toast!";
int duration = Toast.LENGTH_LONG;

Toast toast = Toast.makeText(context, text, duration);
toast.show();
```

lifechanging University of Sunderland

Positioning your Toast

- Default location is the bottom centre of the screen.
- Can be changed with setGravity(int gravity, int x-offset, int y-offset)
- Top Left Corner:

```
toast.setGravity(Gravity.TOP|Gravity.LEFT, 0, 0);
```

lifechanging University of Sunderland

Advanced Toast Operation

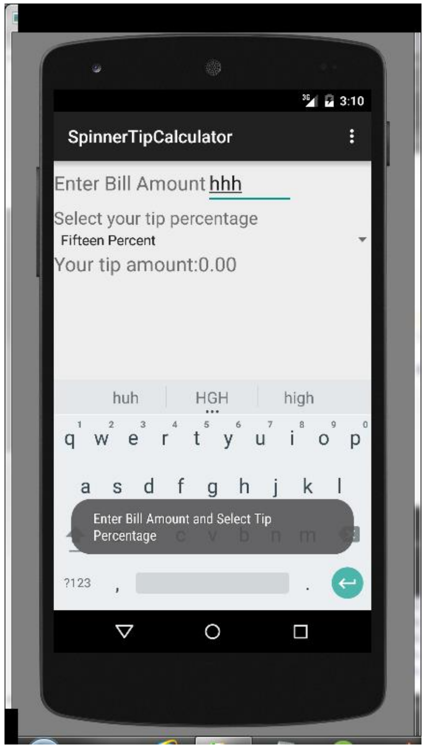
- You can create your own view (layout) to give Toast a customised look.
  - Create layout
  - Use setView() to assign the layout to the toast
  - Inflate the view and display the toast

```
LayoutInflater inflater = getLayoutInflater();
View layout = inflater.inflate(R.layout.custom_toast,
                               (ViewGroup) findViewById(R.id.toast_layout_root));

TextView text = (TextView) layout.findViewById(R.id.text);
text.setText("This is a custom toast");

Toast toast = new Toast(getApplicationContext());
toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(layout);
toast.show();
```

http://developer.android.com/guide/topics/ui/notifiers/toasts.html

Tip Calculator with Toast

```
if(isValidInput){
    //calculate tip amount}
else{
    Context myContext = this.getApplicationContext();
    Toast myToast = new Toast(myContext);
    myToast.makeText(myContext,
    "Enter Bill Amount and Select Tip Percentage",
    Toast.LENGTH_SHORT).show();
    //reset tip amount
    textViewTipAmount.setText("0.00");
}
```


SpinnerTipCalculator
Enter Bill Amount hhh
Select your tip percentage
Fifteen Percent
Your tip amount:0.00
Enter Bill Amount and Select Tip Percentage

Tip Calculator with Toast

```
private boolean checkBillInput(String s) {
    try {
        Integer.parseInt(s);
    } catch(NumberFormatException e) {
        return false;
    } catch(NullPointerException e) {
        return false;
    }
    // only got here if we didn't return false
    return true;
}
```

Pickers

- Allows user to select a date / time from a set range of inputs
  - Reduces validation logic required
- Two Classes:
  - DatePicker
  - TimePicker

- Class for handling date / time information
  - Calendar
  - Date

- Listeners:
  - DatePicker.OnDateChangedListener
  - TimePicker.OnTimeChangedListener

Date Picker

Class: DatePicker

Package: android.widget

Extends: android.widget.FrameLayout

Overview: Widget for date selection. The date is set by a series of Spinners. Date can be selected via the Spinners or via a CalendarView object.

Methods:

Init(int year, int month, int day, DatePicker.OnDateChanged Listener listener)

> This method sets the initial date for the object and
> assigns the listener to the object.

DatePicker.OnDateChanged(Listener listener)

> Overridden method where we implement application logic.

Full API: http://developer.android.com/reference/android/widget/DatePicker.html

Basic Operation

```
public class PickerActivity extends ActionBarActivity
                implements DatePicker.OnDateChangedListener {

    DatePicker myPicker = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        myPicker = (DatePicker)findViewById(R.id.datePicker);
        // initialise the datePicker
        // Otherwise onDateChange events are not picked up
        myPicker.init(myPicker.getYear(), myPicker.getMonth(),
               myPicker.getDayOfMonth(), this);
    }
```

lifechanging University of Sunderland

Basic Operation

- Override onDateChanged() method with the functionality that we want.

```
@Override
public void onDateChanged(DatePicker view, int year,
                          int monthOfYear, int dayOfMonth) {

        // Application logic goes here

}
```

Summary

- Components you're now aware of:
  - TextView
  - EditText
  - Button
  - Spinner
  - Toast
  - CheckBox
  - DatePicker
  - TimePicker

- You should also have an appreciation of how components can be used to help with basic input validation.
  - And how additional application logic can ensure the rest.

- There are lots more components out there - explore the API documentation.

Summary

- There is a general formula for components:
  - Create in XML
    - Assign properties
  - Create variables to represent the components (in your activity Class)
  - Assign Listeners to the components the user will interact with (in your activity Class)
  - Capture the listener events by implementing overridden methods, and implement your application logic.

lifechanging University of Sunderland