<?xml version="1.0" encoding="utf-8"?>
<manifest>
    <uses-permission />
    <uses-sdk />
    <uses-feature />
    <supports-screens />
    <application />
    <activity />
    <intent-filter />
    <service />
    <receiver />
    <uses-library />
</manifest>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.mustafaozcan.tarayici"
    android:versionCode="1"
    android:versionName="1.0">
  <uses-sdk
      android:minSdkVersion="10"
      android:targetSdkVersion="16"/>
  <uses-permission android:name="android.permission INTERNET"/>
  <application
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme">
    <activity
      android:name=".MainActivity"
      android:label="@string/title_activity_main">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>

Android User Interface

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a Button" />
</LinearLayout>

http://developer.android.com/guide/topics/ui/overview.html
**Layouts**

**Linear Layout**
A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

**Relative Layout**
Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).

**WebView**
Displays web pages.

Common layouts backed by an adapter include:

**List View**
Displays a scrolling single column list.

**GridView**
Displays a scrolling grid of columns and rows.

# Android Input Controls

## Common Controls

<table>
<thead>
<tr>
<th>Control Type</th>
<th>Description</th>
<th>Related Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button</td>
<td>A push-button that can be pressed, or clicked, by the user to perform an action.</td>
<td>Button</td>
</tr>
<tr>
<td>Text field</td>
<td>An editable text field. You can use the <strong>AutoCompleteTextView</strong> widget to create a text entry widget that provides auto-complete suggestions</td>
<td>EditText, AutoCompleteTextView</td>
</tr>
<tr>
<td>Checkbox</td>
<td>An on/off switch that can be toggled by the user. You should use checkboxes when presenting users with a group of selectable options that are not mutually exclusive.</td>
<td>CheckBox</td>
</tr>
<tr>
<td>Radio button</td>
<td>Similar to checkboxes, except that only one option can be selected in the group.</td>
<td>RadioGroup, RadioButton</td>
</tr>
<tr>
<td>Toggle button</td>
<td>An on/off button with a light indicator.</td>
<td>ToggleButton</td>
</tr>
<tr>
<td>Spinner</td>
<td>A drop-down list that allows users to select one value from a set.</td>
<td>Spinner</td>
</tr>
<tr>
<td>Pickers</td>
<td>A dialog for users to select a single value for a set by using up/down buttons or via a swipe gesture. Use a <strong>DatePicker</strong> or a <strong>TimePicker</strong> widget to enter the values for the date (month, day, year) or a <strong>TimePicker</strong> widget to enter the values for a time (hour, minute, AM/PM), which will be formatted automatically for the user’s locale.</td>
<td>DatePicker, TimePicker</td>
</tr>
</tbody>
</table>

**Event Listeners**

**onClickListener()**
From `View.OnClickListener`. This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.

**onLongClickListener()**
From `View.OnLongClickListener`. This is called when the user either touches and holds the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses and holds the suitable "enter" key or presses and holds down on the trackball (for one second).

**onFocusChangeListener()**
From `View.OnFocusChangeListener`. This is called when the user navigates onto or away from the item, using the navigation-keys or trackball.

**onKey()**
From `View.OnKeyListener`. This is called when the user is focused on the item and presses or releases a hardware key on the device.

**onTouch()**
From `View.OnTouchListener`. This is called when the user performs an action qualified as a touch event, including a press, a release, or any movement gesture on the screen (within the bounds of the item).

**onCreateContextMenu()**
From `View.onCreateContextMenuListener`. This is called when a Context Menu is being built (as the result of a sustained "long click"). See the discussion on context menus in the Menus developer guide.

Menus

Options Menu

Context Menu

Dialogs

Notifications

Figure 1. Notifications in the notification area.

Figure 2. Notifications in the notification drawer.

Figure 3. Notification in normal view.

The callouts in the illustration refer to the following:

1. Content title
2. Large icon
3. Content text
4. Content info
5. Small icon
6. Time that the notification was generated to the time that the system received it

Figure 4. Big view notification.

Toasts

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

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

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

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

http://developer.android.com/guide/components/activities.html
Activity Lifecycle

- onCreate() or onRestoreInstanceState()
  - Restore your activity state

- User navigates to the activity

- App process killed*
  - Apps with higher priority need memory

- Activity is not visible
  - Activity instance is intact; no need to restore the state
  - onRestart()
  - User navigates to the activity

- onSaveInstanceState()
  - Save your activity state

- Another activity comes into the foreground

*Activity instance is destroyed, but the state from onSaveInstanceState() is saved

http://developer.android.com/guide/components/activities.html
Rubric

Creating a Rubric for Projects

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Poor</th>
<th>Average</th>
<th>Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Working Correctly</td>
<td>10</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Usability</td>
<td>5</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Standardization</td>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Delivery On Time</td>
<td>5</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>
Assesment

• Ara Sınav (Proje Ödevi)
• Ölçme faaliyeti (En az 2)
• Final

Projects

• 30.10.2012
• 13.11.2012
• 18.12.2012
• 25.12.2012

Exams

• Ara Sınav 20.11.2012 – 04.12.2012 Project

Calculator

**UI Elements:**
- Linear / Relative / Table Layout
- EditText
- Buttons

**Scenario:**
- Basic Math Operations
Mobile Browser

UI Elements:
• Linear Layout
• EditText
• Button
• WebView

Scenario:
• Write url
• Click to Go Button
• Show Toast Notification (URL)
• Load url in WebView
• Set Javascript enable for WebView
<Thanks />

www.mustafaozcan.net

ozcanmustafa@gmail.com