**Experiment No. 6**

PART A

(PART A: TO BE REFFERED BY STUDENTS)

A.1 Aim: **User Interface Design Phase 3**

**A.2 Prerequisite:**

Knowledge of java and android app.

**A.3 Outcome:**

After successful completion of this experiment students will be able to

1. Make screen layout for android app.
2. Know the idea of sdk manager and AVM.
3. Understand the uses of tools available for user interface.

**A.4 Theory:**

Android UI Controls

There are number of UI controls provided by Android that allow you to build the graphical user interface for your app.

|  |  |
| --- | --- |
| S.N. | UI Control & Description |
| 1 | [TextView](http://www.tutorialspoint.com/android/android_textview_control.htm) This control is used to display text to the user. |
| 2 | [EditText](http://www.tutorialspoint.com/android/android_edittext_control.htm) EditText is a predefined subclass of TextView that includes rich editing capabilities. |
| 3 | [AutoCompleteTextView](http://www.tutorialspoint.com/android/android_autocompletetextview_control.htm) The AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing. |
| 4 | [Button](http://www.tutorialspoint.com/android/android_button_control.htm) A push-button that can be pressed, or clicked, by the user to perform an action. |
| 5 | [ImageButton](http://www.tutorialspoint.com/android/android_imagebutton_control.htm) AbsoluteLayout enables you to specify the exact location of its children. |
| 6 | [CheckBox](http://www.tutorialspoint.com/android/android_checkbox_control.htm) 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. |
| 7 | [ToggleButton](http://www.tutorialspoint.com/android/android_togglebutton_control.htm) An on/off button with a light indicator. |
| 8 | [RadioButton](http://www.tutorialspoint.com/android/android_radiobutton_control.htm) The RadioButton has two states: either checked or unchecked. |
| 9 | [RadioGroup](http://www.tutorialspoint.com/android/android_radiogroup_control.htm) A RadioGroup is used to group together one or more RadioButtons. |
| 10 | [ProgressBar](http://www.tutorialspoint.com/android/android_progressbar.htm) The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background. |
| 11 | [Spinner](http://www.tutorialspoint.com/android/android_spinner_control.htm) A drop-down list that allows users to select one value from a set. |
| 12 | [TimePicker](http://www.tutorialspoint.com/android/android_timepicker_control.htm) The TimePicker view enables users to select a time of the day, in either 24-hour mode or AM/PM mode. |
| 13 | [DatePicker](http://www.tutorialspoint.com/android/android_datepicker_control.htm) The DatePicker view enables users to select a date of the day. |

**Create UI Controls**

A view object have a unique ID assigned to it which will identify the View uniquely within the tree. The syntax for an ID, inside an XML tag is:

**android:id="@+id/text\_id"**

To create a UI Control/View/Widget we will have to define a view/widget in the layout file and assign it a unique ID as follows:

<?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\_id"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="I am a TextView" />

</LinearLayout>

Then finally create an instance of the Control object and capture it from the layout, use the following:

TextView myText = (TextView) findViewById(R.id.text\_id);

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

**(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)**

|  |  |
| --- | --- |
| Roll No. B046 | Name: Pranav Joshi |
| Program: B.tech Computer | Division: B |
| Semester: IV | Batch : B3 |
| Date of Experiment: 5-3-15 | Date of Submission: 19-3-15 |
| Grade : |  |

B.1 Task to be done:

**1)** Paste Source code of .xml and .java files in this section along with output.

**<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:tools="http://schemas.android.com/tools"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**android:background="@android:color/black"**

**>**

**<ImageView**

**android:id="@+id/image"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:src="@drawable/wine"**

**android:layout\_centerInParent="true"**

**/>**

**</RelativeLayout>**

**JAVA CODE:**

package com.drunktest;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

import android.support.v7.app.ActionBarActivity;

import android.view.Window;

import android.view.animation.AccelerateInterpolator;

import android.view.animation.AlphaAnimation;

import android.view.animation.Animation;

import android.view.animation.AnimationSet;

import android.view.animation.DecelerateInterpolator;

import android.widget.ImageView;

public class SplashActivity extends ActionBarActivity {

private final Handler handler = new Handler();

private ImageView img;

private final Runnable startActivityRunnable = new Runnable() {

@Override

public void run() {

Intent intent = new Intent();

intent.setClass(SplashActivity.this,HomeActivity.class);

startActivity(intent);

finish();

}

};

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

setContentView(R.layout.splash);

img = (ImageView) findViewById (R.id.image);

//setContentView(img);

}

@Override

protected void onResume() {

super.onResume();

Animation fadeIn = new AlphaAnimation(0, 1);

fadeIn.setInterpolator(new DecelerateInterpolator()); //add this

fadeIn.setDuration(2000);

Animation fadeOut = new AlphaAnimation(1, 0);

fadeOut.setInterpolator(new AccelerateInterpolator()); //and this

fadeOut.setStartOffset(2500);

fadeOut.setDuration(2000);

AnimationSet animation = new AnimationSet(false); //change to false

animation.addAnimation(fadeIn);

animation.addAnimation(fadeOut);

animation.setFillAfter(true);

img.startAnimation(animation);

handler.postDelayed(startActivityRunnable, 5000);

}

public void onPause()

{

super.onPause();

handler.removeCallbacks(startActivityRunnable);

}

@Override

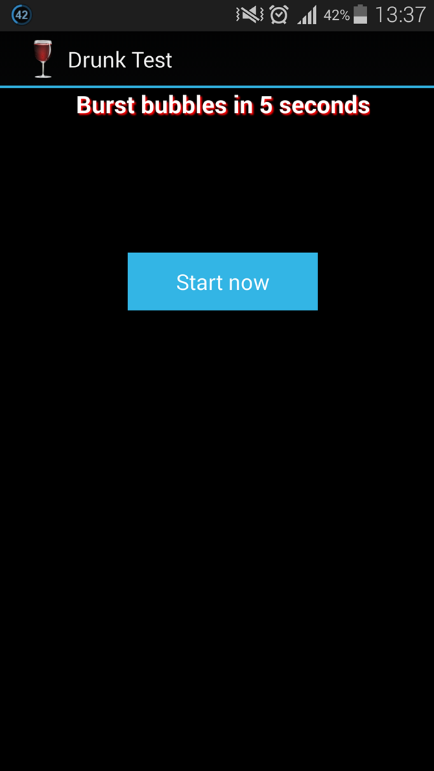
public void onBackPressed() {

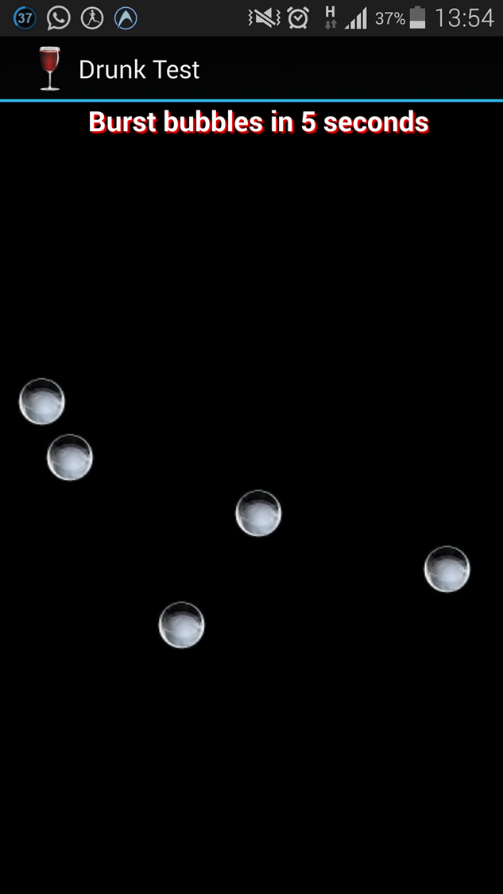
// TODO Auto-generated method stub

//super.onBackPressed();

}

}





B.3 Conclusion:

Hence we used java coding and xml resources to solidify our progress with respect to the application.

We successfully used GUI development assisting java classes and combined other class resources to further practical demonstration of our application.