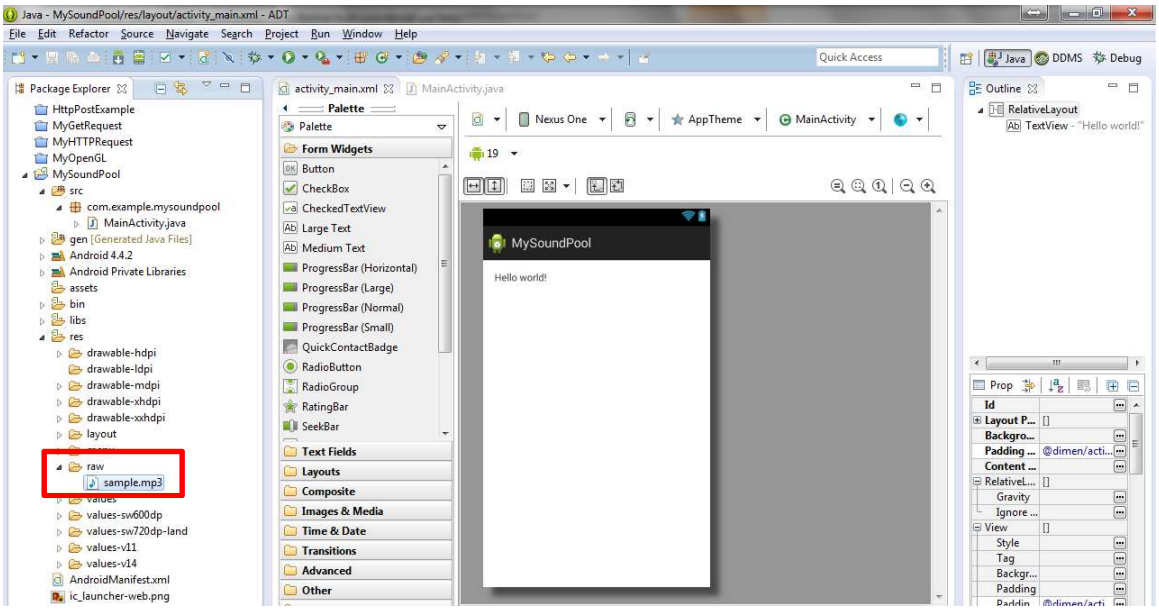
1. Audio Player

1.1 Sound Pool

1. Create the Android application with the following attributes.
   - Application Name: **MySoundPool**
   - Project Name: **MySoundPool**
   - Package Name: **com.example.mysoundpool**

2. Create a folder "**raw**" under "**res**" and drag the audio "**sample.mp3**" into the folder.

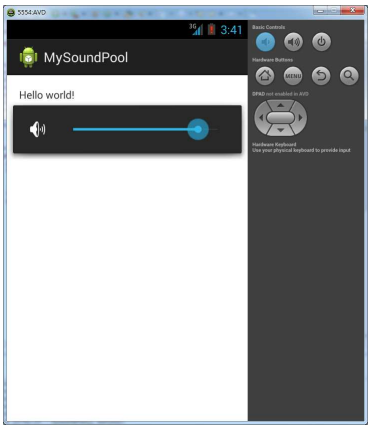


3. Modify the source file "**MainActivity.java**" as follow:

```
package com.example.mysoundpool;


import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.media.AudioManager;
import android.media.SoundPool;
import android.media.SoundPool.OnLoadCompleteListener;


public class MainActivity extends Activity {
    private SoundPool soundPool;
    private int soundID;
    private float volume;
```

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

        // Set the hardware buttons to control the music
        this.setVolumeControlStream(AudioManager.STREAM_MUSIC);

        // Getting the user sound settings
        AudioManager audioManager = (AudioManager) getSystemService(AUDIO_SERVICE);
        float actualVolume = (float)
            audioManager.getStreamVolume(AudioManager.STREAM_MUSIC);
        float maxVolume = (float)
            audioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC);

        // Define the volume
        volume = actualVolume / maxVolume;

        // Define the media clip to be loaded
        soundID = soundPool.load(this, R.raw.sample, 1);

        // Load the sound
        soundPool = new SoundPool(10, AudioManager.STREAM_MUSIC, 0);
        soundPool.setOnLoadCompleteListener(new OnLoadCompleteListener() {
            @Override
            public void onLoadComplete(SoundPool soundPool, int sampleId,
                                       int status) {
                soundPool.play(soundID, volume, volume, 1, 0, 1f);
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

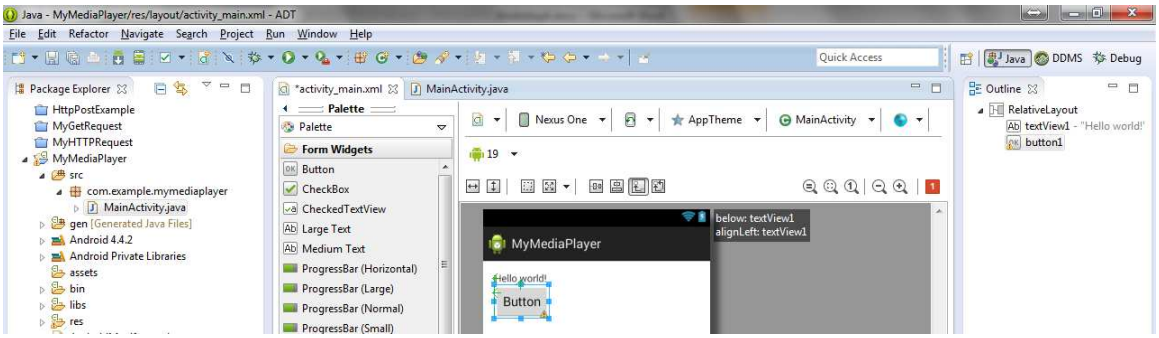
4. Turn on your speaker and then execute the app, can you hear the beautiful song? Press the volume button to adjust the sound.



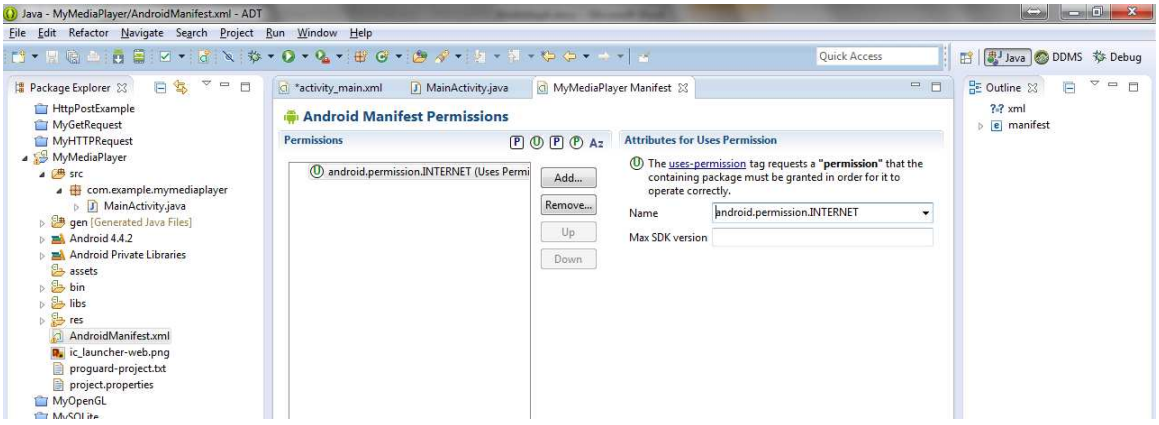
1.2 Media Player

1. Create the Android application with the following attributes.
   - Application Name: **MyMediaPlayer**
   - Project Name: **MyMediaPlayer**
   - Package Name: **com.example.mymediaplayer**

2. Drag a button to the layout.



3. Add the following user permission in "**AndroidManifest.xml**":
   - android.permission.INTERNET

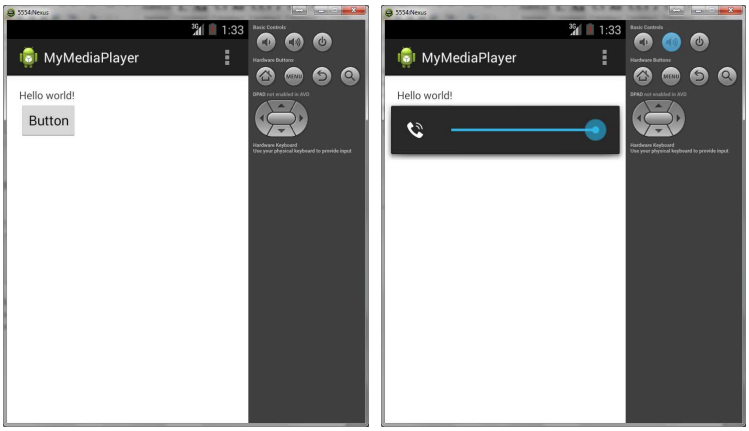


4. Modify the source file "**MainActivity.java**" as follow:

```
package com.example.mymediaplayer;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.media.AudioManager;
import android.media.MediaPlayer;

public class MainActivity extends Activity {
    Button mButtonStop;
    MediaPlayer mPlayer;
    String url = "http://www.peter-lo.com/Teaching/4T025-2-A/sample.mp3";

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

        try {
            // Create a new media player
            mPlayer = new MediaPlayer();

            // Sets the audio stream type for this MediaPlayer.
            mPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);

            // Get the source from URL
            mPlayer.setDataSource(url);

            // Prepare the player
            mPlayer.prepare();

            // Start playing music
            mPlayer.start();
        } catch (Exception e) {
            e.printStackTrace();
        }

        // Create the listener for the Stop button
```

```
        mButtonStop = (Button) findViewById(R.id.button1);
        mButtonStop.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                if(mPlayer!=null && mPlayer.isPlaying()){
                    mPlayer.stop();
                }
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

5. Turn on your speaker and then execute the app, can you hear the beautiful song? Press the volume button to adjust the sound. This app can continue playing the music even minimize it, you need to press the button to stop the music.



2. Audio Recorder

2.1 Simple Audio Recorder

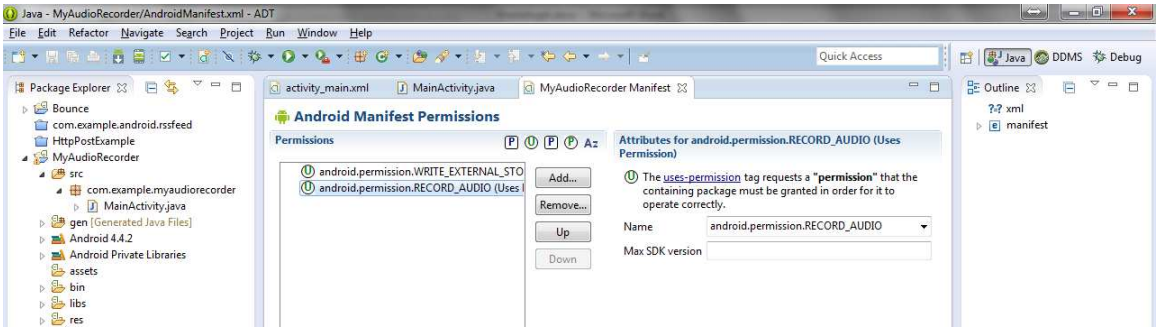
1. Create the Android application with the following attributes.
   - Application Name: **MyAudioRecorder**
   - Project Name: **MyAudioRecorder**
   - Package Name: **com.example.myaudiorecorder**

2. Modify the source file "**MainActivity.java**" as follow:

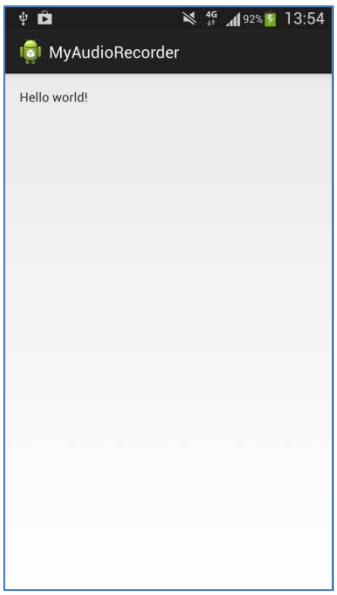
```
package com.example.myaudiorecorder;


import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Intent;
import android.media.MediaRecorder;
import android.net.Uri;
import android.os.Environment;
import android.provider.MediaStore;
import java.io.File;


public class MainActivity extends Activity {
    private MediaRecorder mRecorder;
    private File audiofile = null;

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

        File sampleDir = Environment.getExternalStorageDirectory();
        try {
            audiofile = File.createTempFile("sound", ".3gp", sampleDir);
            mRecorder = new MediaRecorder();
            mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
            mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
```

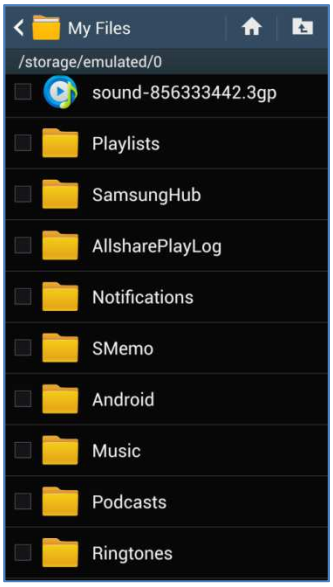
```
            mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
            mRecorder.setOutputFile(audiofile.getAbsolutePath());
            mRecorder.prepare();
            mRecorder.start();
        } catch (Exception e) {
        }
    }

    @Override
    protected void onPause() {
        super.onPause();

        // Stop recording and release resource
        mRecorder.stop();
        mRecorder.release();

        // Construct the content value
        ContentValues values = new ContentValues(4);
        values.put(MediaStore.Audio.Media.TITLE, "audio" + audiofile.getName());
        values.put(MediaStore.Audio.Media.DATE_ADDED,
                (int) (System.currentTimeMillis() / 1000));
        values.put(MediaStore.Audio.Media.MIME_TYPE, "audio/3gpp");
        values.put(MediaStore.Audio.Media.DATA, audiofile.getAbsolutePath());

        ContentResolver contentResolver = getContentResolver();
        Uri newUri = contentResolver.insert(
                MediaStore.Audio.Media.EXTERNAL_CONTENT_URI, values);

        // Broadcast the given intent to all interested BroadcastReceivers
        sendBroadcast(new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE, newUri));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

3. Add the following user permission in "**AndroidManifest.xml**":
   - android.permission.WRITE_EXTERNAL_STORAGE
   - android.permission.RECORD_AUDIO



4. Save and execute the app in your mobile. You can start record your voice once the app start, and stop it when you close or exit the app. If you want to open your record voice, you can find them in “My Files”. (This app is not workable in emulator)



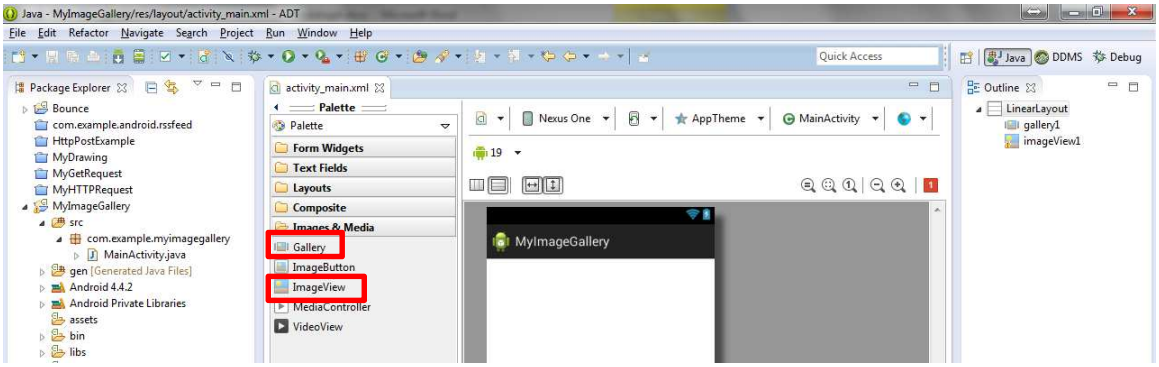


3. Photo Album

3.1 Image Gallery

1. Create the Android application with the following attributes.
   - Application Name: **MyImageGallery**
   - Project Name: **MyImageGallery**
   - Package Name: **com.example.myimagegallery**

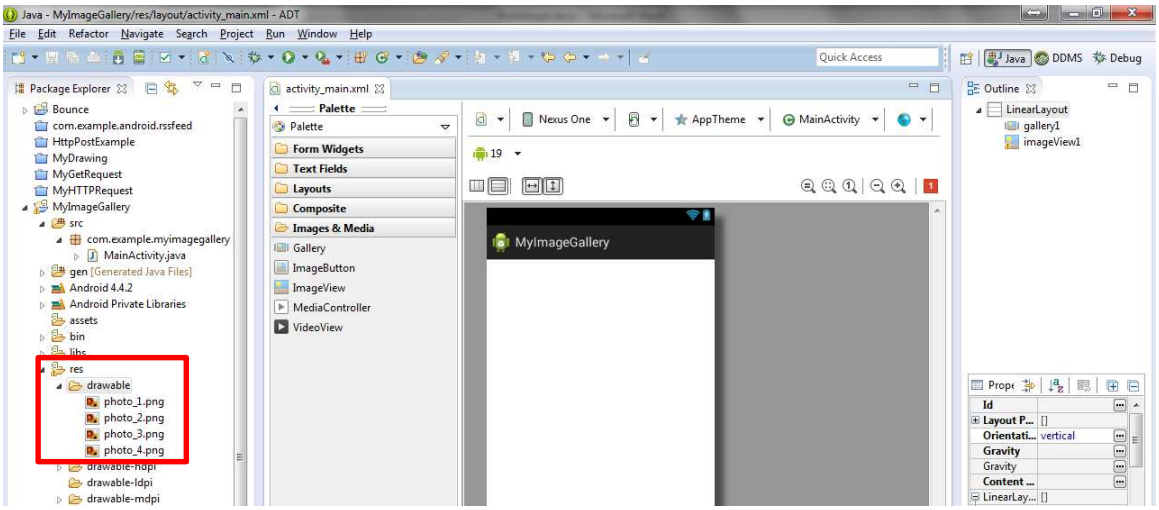
2. Remove the default text view “Hello World”. Then then add an ImageGallery and ImageView to the layout.



3. The XML code for the layout should like:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

<Gallery
    android:id="@+id/gallery1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

<ImageView
    android:id="@+id/imageView1"
    android:layout_gravity="center_vertical|center_horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

4. Select the folder "**res**", then right click and select **New ➔ Folder** to create a new folder "**drawable**". Then put the pictures "photo_1.png", "photo_2.png", "photo_3.png" and "photo_4.png" in it.



5. Select the folder "**res/value**", then right click and select **New ➔ Android XML File** to create the file "**attrs.xml**". Then modify the content as follow:

```
<resources>
    <declare-styleable name="HelloGallery">
        <attr name="android:galleryItemBackground" />
    </declare-styleable>
</resources>
```

6. Modify the source file "**MainActivity.java**" as follow:

```
package com.example.myimagegallery;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Context;
import android.content.res.TypedArray;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
```

```
public class MainActivity extends Activity {
    private Integer[] mImageIds = {
            R.drawable.photo_1,
            R.drawable.photo_2,
            R.drawable.photo_3,
            R.drawable.photo_4,
    };

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

        Gallery g = (Gallery) findViewById(R.id.gallery1);
        g.setAdapter(new ImageAdapter(this));
        g.setOnItemClickListener(new OnItemClickListener() {
            public void onItemClick(
              @SuppressWarnings("rawtypes")
              AdapterView parent, View v, int position, long id) {
              ImageView imageview = (ImageView) findViewById(R.id.imageView1);
              imageview.setImageResource(mImageIds[position]);
            }
        });
    }

    public class ImageAdapter extends BaseAdapter {
        int mGalleryItemBackground;
        private Context mContext;

        public ImageAdapter(Context c) {
            mContext = c;
            TypedArray a = obtainStyledAttributes(R.styleable.HelloGallery);
            mGalleryItemBackground = a.getResourceId(
                    R.styleable.HelloGallery_android_galleryItemBackground, 0);
            a.recycle();
        }

        public int getCount() {
            return mImageIds.length;
```

```
        }

        public Object getItem(int position) {
            return position;
        }

        public long getItemId(int position) {
            return position;
        }

        public View getView(int position, View convertView, ViewGroup parent) {
            ImageView i = new ImageView(mContext);
            i.setImageResource(mImageIds[position]);
            i.setLayoutParams(new Gallery.LayoutParams(150, 100));
            i.setScaleType(ImageView.ScaleType.FIT_XY);
            i.setBackgroundResource(mGalleryItemBackground);
            return i;
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

7. Save and execute your app. You can display the photo after you scroll and select the image.

4. Graphics

4.1 Create a Simple Drawing

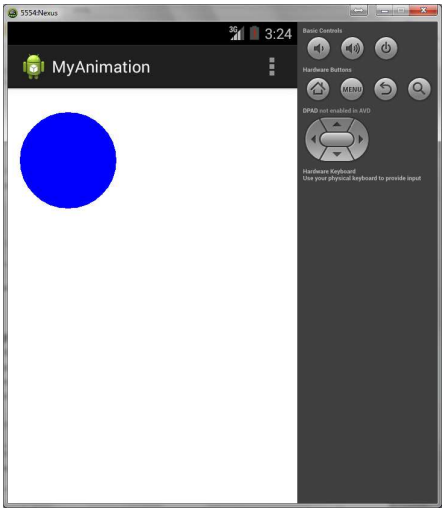
1. Create the Android application with the following attributes.
   - Application Name: **MyDrawing**
   - Project Name: **MyDrawing**
   - Package Name: **com.example.mydrawing**

2. Modify the source file "**MainActivity.java**" as follow:

```
package com.example.mydrawing;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//      setContentView(R.layout.activity_main);
        View MyView = new MyView(this);
        setContentView(MyView);
    }

    class MyView extends View {
        private float Radius = 80;
        private float X = 100;
        private float Y = 120;
        private RectF mBall;   // Needed for Canvas.drawOval
        private Paint mPaint;  // The paint (e.g. style, color) used for drawing

        // Constructor
```

```
        public MyView(Context context) {
            super(context);

            // Needed for Canvas.drawOval
            mBall = new RectF();

            // The paint (e.g. style, color) used for drawing
            mPaint = new Paint();
        }

        @Override
        protected void onDraw(Canvas canvas) {
            // Draw the ball
            mBall.set(X - Radius, Y - Radius, X + Radius, Y + Radius);
            mPaint.setColor(Color.BLUE);
            canvas.drawOval(mBall, mPaint);
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

3. Save and execute the app, you should able to see a blue circle on the screen.



4.2 Create a Simple Animation

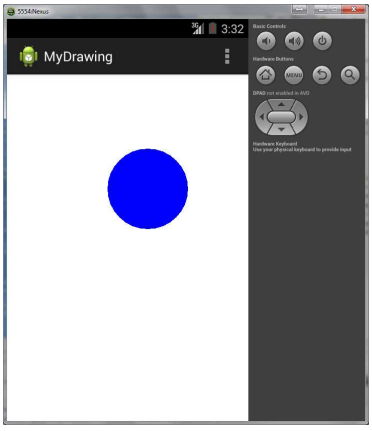
1. Modify the source file "**MainActivity.java**" as follow:

```
package com.example.mydrawing;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//      setContentView(R.layout.activity_main);
        View MyView = new MyView(this);
        setContentView(MyView);
    }

    class MyView extends View {
        private float Radius = 80;
        private float X = 100;
        private float Y = 120;
        private RectF mBall;
        private Paint mPaint;
        private int xMin = 0;
        private int xMax;
        private int yMin = 0;
        private int yMax;
        private float xSpeed = 5;
        private float ySpeed = 3;

        // Constructor
        public MyView(Context context) {
            super(context);
```

```
        // Needed for Canvas.drawOval
        mBall = new RectF();

        // The paint (e.g. style, color) used for drawing
        mPaint = new Paint();
    }

    @Override
    protected void onDraw(Canvas canvas) {
        // Draw the ball
        mBall.set(X - Radius, Y - Radius, X + Radius, Y + Radius);
        mPaint.setColor(Color.BLUE);
        canvas.drawOval(mBall, mPaint);

        // Update the position, including collision detection and reaction
        update();

        // Delay
        try {
            Thread.sleep(30);
        } catch (Exception e) {
        }

        // Force to call OnDraw() again for re-draw if invalidate occur
        invalidate();
    }

    // Detect collision and update the position of the ball.
    private void update() {
        // Get new (x,y) position
        X += xSpeed;
        Y += ySpeed;

        // Detect collision and react
        if (X + Radius > xMax) {
            xSpeed = -xSpeed;
            X = xMax - Radius;
        } else if (X - Radius < xMin) {
```

```
                    xSpeed = -xSpeed;
                    X = xMin + Radius;
               }
               if (Y + Radius > yMax) {
                    ySpeed = -ySpeed;
                    Y = yMax - Radius;
               } else if (Y - Radius < yMin) {
                    ySpeed = -ySpeed;
                    Y = yMin + Radius;
               }
          }

          @Override
          public void onSizeChanged(int w, int h, int oldW, int oldH) {
               // Called back when the view is first created or its size changes.
               // Set the movement bounds for the ball
               xMax = w - 1;
               yMax = h - 1;
          }
     }

     @Override
     public boolean onCreateOptionsMenu(Menu menu) {
          // Inflate the menu; this adds items to the action bar if it is present.
          getMenuInflater().inflate(R.menu.main, menu);
          return true;
     }
}
```

2. Save and execute the app, can you see the blue ball moving around the screen?



5. Open GL ES

5.1 Simple 3D Cube

1. Create the Android application with the following attributes.
   - Application Name: **MySimpleCube**
   - Project Name: **MySimpleCube**
   - Package Name: **com.example.mysimplecube**

2. Modify the source file "**MainActivity.java**" as follow:

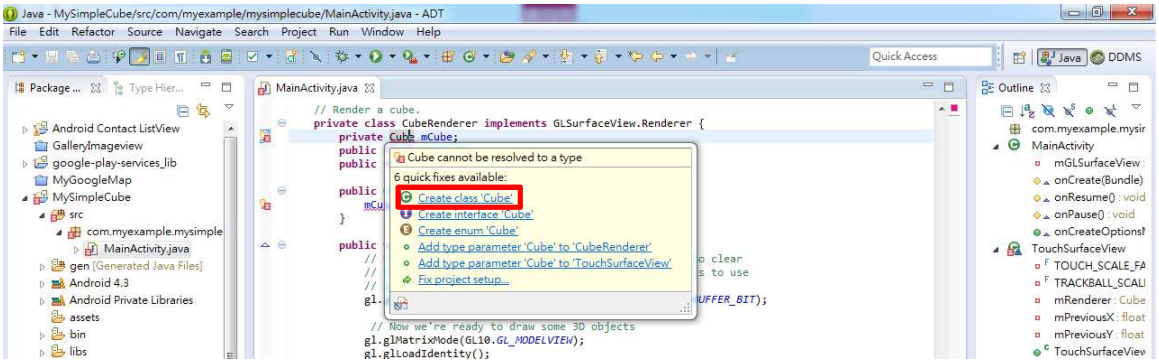
```
package com.example.mysimplecube;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;
import android.content.Context;
import android.opengl.GLSurfaceView;
import android.view.MotionEvent;

public class MainActivity extends Activity {
    private GLSurfaceView mGLSurfaceView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//      setContentView(R.layout.activity_main);
        // Create our Preview view and set it as the content of our Activity
        mGLSurfaceView = new TouchSurfaceView(this);
        setContentView(mGLSurfaceView);
        mGLSurfaceView.requestFocus();
        mGLSurfaceView.setFocusableInTouchMode(true);
    }

    @Override
    protected void onResume() {
        // Ideally a game should implement onResume() and onPause() to take appropriate
        // action when the activity looses focus
```

```
        super.onResume();
        mGLSurfaceView.onResume();
    }

    @Override
    protected void onPause() {
        // Ideally a game should implement onResume() and onPause() to take appropriate
        // action when the activity looses focus
        super.onPause();
        mGLSurfaceView.onPause();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}

// Implement a simple rotation control.
class TouchSurfaceView extends GLSurfaceView {
    private final float TOUCH_SCALE_FACTOR = 180.0f / 320;
    private final float TRACKBALL_SCALE_FACTOR = 36.0f;
    private CubeRenderer mRenderer;
    private float mPreviousX;
    private float mPreviousY;

    public TouchSurfaceView(Context context) {
        super(context);
        mRenderer = new CubeRenderer();
        setRenderer(mRenderer);
        setRenderMode(GLSurfaceView.RENDERMODE_WHEN_DIRTY);
    }

    @Override
    public boolean onTrackballEvent(MotionEvent e) {
        mRenderer.mAngleX += e.getX() * TRACKBALL_SCALE_FACTOR;
        mRenderer.mAngleY += e.getY() * TRACKBALL_SCALE_FACTOR;
```

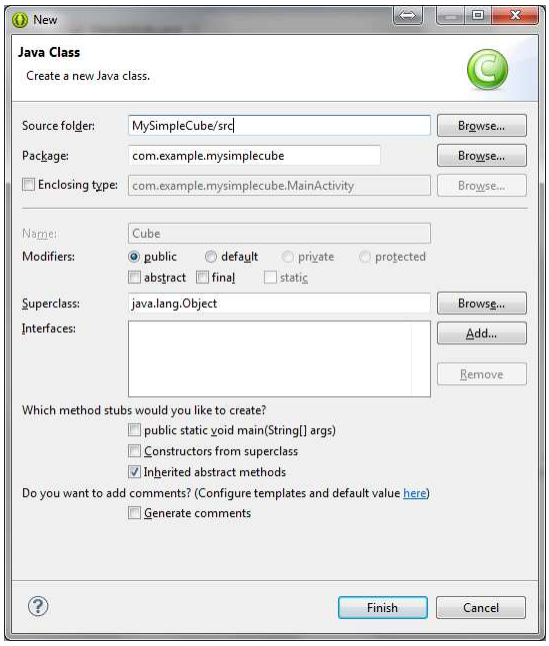
```
        requestRender();
        return true;
    }

    @Override
    public boolean onTouchEvent(MotionEvent e) {
        float x = e.getX();
        float y = e.getY();
        switch (e.getAction()) {
        case MotionEvent.ACTION_MOVE:
            float dx = x - mPreviousX;
            float dy = y - mPreviousY;
            mRenderer.mAngleX += dx * TOUCH_SCALE_FACTOR;
            mRenderer.mAngleY += dy * TOUCH_SCALE_FACTOR;
            requestRender();
        }
        mPreviousX = x;
        mPreviousY = y;
        return true;
    }

    // Render a cube.
    private class CubeRenderer implements GLSurfaceView.Renderer {
         private Cube mCube;
         public float mAngleX;
         public float mAngleY;

         public CubeRenderer() {
              mCube = new Cube();
         }

         public void onDrawFrame(GL10 gl) {
            // Usually, the first thing one might want to do is to clear the screen.
            // The most efficient way of doing this is to use glClear().
            gl.glClear(GL10.GL_COLOR_BUFFER_BIT | GL10.GL_DEPTH_BUFFER_BIT);

             // Now we're ready to draw some 3D objects
            gl.glMatrixMode(GL10.GL_MODELVIEW);
            gl.glLoadIdentity();
```

```
            gl.glTranslatef(0, 0, -3.0f);
            gl.glRotatef(mAngleX, 0, 1, 0);
            gl.glRotatef(mAngleY, 1, 0, 0);

            gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
            gl.glEnableClientState(GL10.GL_COLOR_ARRAY);

            mCube.draw(gl);
        }

        public void onSurfaceChanged(GL10 gl, int width, int height) {
             gl.glViewport(0, 0, width, height);

             // Set our projection matrix. This doesn't have to be done each time we
             // draw, but usually a new projection needs to be set when the viewport
             // is resized.
             float ratio = (float) width / height;
             gl.glMatrixMode(GL10.GL_PROJECTION);
             gl.glLoadIdentity();
             gl.glFrustumf(-ratio, ratio, -1, 1, 1, 10);
        }

        public void onSurfaceCreated(GL10 gl, EGLConfig config) {
             // By default, OpenGL enables features that improve quality but reduce
             // performance. One might want to tweak that especially on software
             // renderer.
             gl.glDisable(GL10.GL_DITHER);

             // Some one-time OpenGL initialization can be made here probably based
             // on features of this particular context
             gl.glHint(GL10.GL_PERSPECTIVE_CORRECTION_HINT, GL10.GL_FASTEST);

             gl.glClearColor(1,1,1,1);
             gl.glEnable(GL10.GL_CULL_FACE);
             gl.glShadeModel(GL10.GL_SMOOTH);
             gl.glEnable(GL10.GL_DEPTH_TEST);
        }
    }
}
```

3. Create the class “Cube”.



4. Input “**Cube**” as the class name and press **[Finish]** button.



5. Modify the source file “**Cube.java**” as follow:

```
package com.example.mysimplecube;

import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.IntBuffer;
import javax.microedition.khronos.opengles.GL10;

public class Cube {
    private IntBuffer   mVertexBuffer;
    private IntBuffer   mColorBuffer;
    private ByteBuffer  mIndexBuffer;
```

```
    public Cube() {
        int one = 0x10000;
        int vertices[] = {
              -one, -one, -one,
               one, -one, -one,
               one,  one, -one,
              -one,  one, -one,
              -one, -one,  one,
               one, -one,  one,
               one,  one,  one,
              -one,  one,  one,
        };

        int colors[] = {
                0,    0,   0,   one,
                one,  0,   0,   one,
                one,  one, 0,   one,
                0,    one, 0,   one,
                0,    0,   one, one,
                one,  0,   one, one,
                one,  one, one, one,
                0,    one, one, one,
        };

        byte indices[] = {
                0, 4, 5,    0, 5, 1,
                1, 5, 6,    1, 6, 2,
                2, 6, 7,    2, 7, 3,
                3, 7, 4,    3, 4, 0,
                4, 7, 6,    4, 6, 5,
                3, 0, 1,    3, 1, 2
        };

         // Buffers to be passed to gl*Pointer() functions must be direct,
         // i.e., they must be placed on the native heap where the garbage collector
         // cannot move them.
         // Buffers with multi-byte datatypes (e.g., short, int, float) must have their
         // byte order set to native order
         ByteBuffer vbb = ByteBuffer.allocateDirect(vertices.length*4);
```

```
            vbb.order(ByteOrder.nativeOrder());
            mVertexBuffer = vbb.asIntBuffer();
            mVertexBuffer.put(vertices);
            mVertexBuffer.position(0);

            ByteBuffer cbb = ByteBuffer.allocateDirect(colors.length*4);
            cbb.order(ByteOrder.nativeOrder());
            mColorBuffer = cbb.asIntBuffer();
            mColorBuffer.put(colors);
            mColorBuffer.position(0);

            mIndexBuffer = ByteBuffer.allocateDirect(indices.length);
            mIndexBuffer.put(indices);
            mIndexBuffer.position(0);
        }

        public void draw(GL10 gl) {
            gl.glFrontFace(gl.GL_CW);
            gl.glVertexPointer(3, gl.GL_FIXED, 0, mVertexBuffer);
            gl.glColorPointer(4, gl.GL_FIXED, 0, mColorBuffer);
            gl.glDrawElements(gl.GL_TRIANGLES, 36, gl.GL_UNSIGNED_BYTE, mIndexBuffer);
        }
    }
```

6. Save and execute the app.