Mobile Computing

Drawing and Animation of Graphics

Custom Drawing

- Every view component in Android can be customized by overriding the relevant drawing functions available in the particular view class.
- Each time a view component is redrawn according to the instructions written inside a method named as onDraw().
- In this lesson we use the ImageView to draw shapes and animation.

Modification to the Manifest

- Drawing the entire canvas as a custom work will require to handle the orientation.
  - Can be done modifying the Android Manifest

```
<activity android:screenOrientation="portrait"
      android:configChanges="orientation|keyboardHidden"
android:name=".MainActivity">
```

Layout

- The custom view component can be added using the layout xml.

```
<LinearLayout
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:orientation="vertical"
    android:background="#000000"
    tools:context="com.example.ravimal.drawingcanvas.MainActivity">

    <com.example.ravimal.drawingcanvas.AnimatedView
        android:id="@+id/anim_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Note that we still have not created the AnimatedView Class.

AnimatedView Class

- This class is used to modify the image view component for custom drawing.
- The class is extended to the android class ImageView and override the method “**protected void** onDraw(Canvas canvas)”
- For animations we can refresh the view rapidly using a handler.
- The canvas object can be used to alter the new drawings to the display

```
public class AnimatedView extends ImageView {

    private Context mContext;
    private Handler h;
    int x=0,y=o;

    public AnimatedView(Context context, AttributeSet attrs) {
        super(context, attrs);
        mContext = context;
        h= new Handler();
    }


    private Runnable r = new Runnable() {
        @Override
        public void run() {
            invalidate();
        }
    };

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        Paint p = new Paint();
        p.setColor(Color.parseColor("#FF00F0"));
        canvas.drawRect(x, y, x+100, y+100, p);
         x++;y++;
        h.postDelayed(r, 30);

    }
}
```

Exercise

- Create an animation of a moving ball from left side to the right side of the screen.
- Change the color of the ball randomly while in the motion
- Add an acceleration for the movement.

Exercise 2

- Create an animation of a moving ball that bounce back when hit the edges of the screen.
- Use the OnTouchListener to get the touch inputs and relocate the ball to the touch position.