Arrays of Buttons

Inside Android

The Complete Code Listing.

Be careful about cutting and pasting.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Great Title!"
        android:textSize="30sp"/>

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_gravity="center"
        android:layout_height="match_parent"
        android:rowCount="5"
        android:columnCount="4"
        android:id="@+id/mygrid">
    </GridLayout>

</LinearLayout>
```

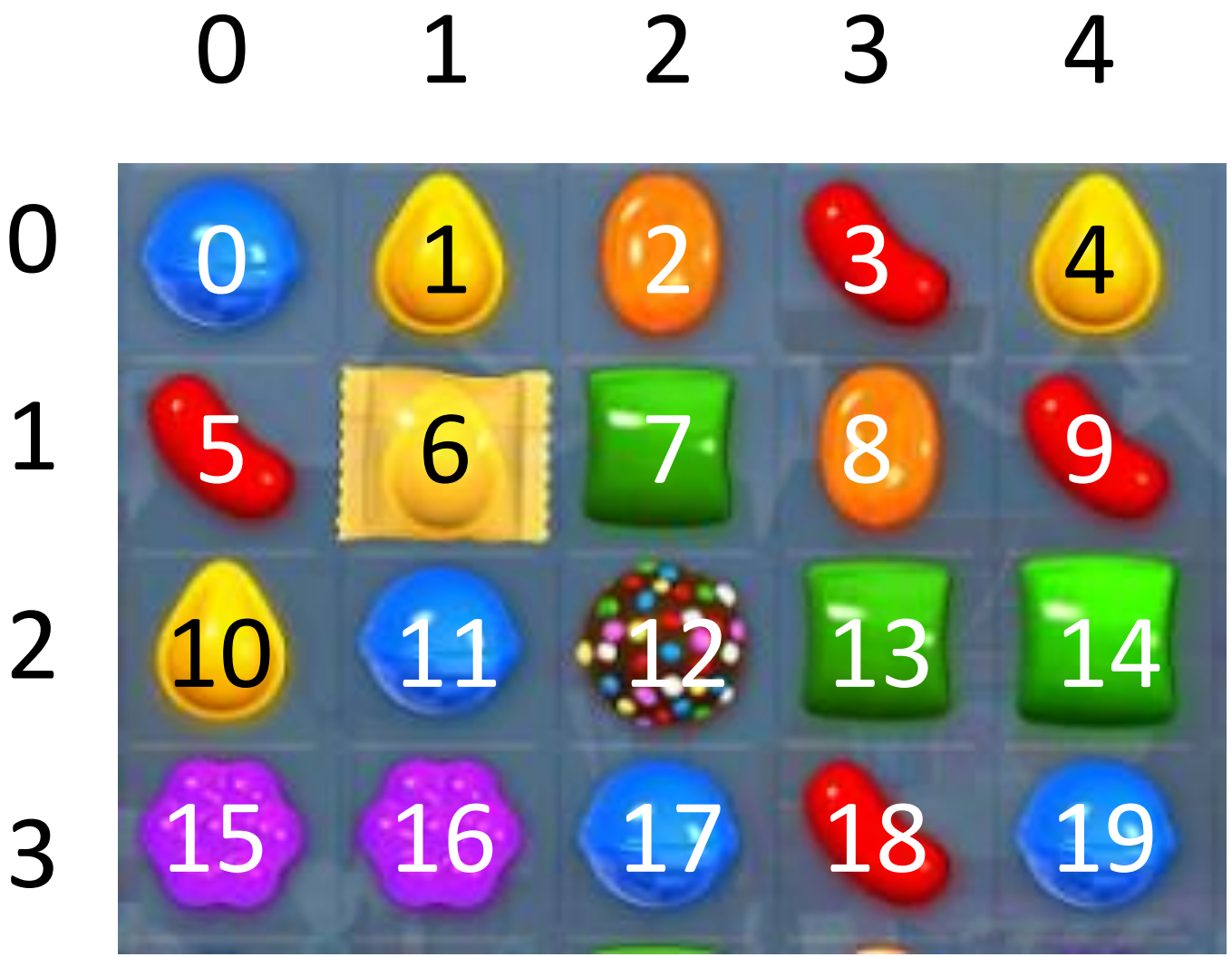
```
public class Game extends AppCompatActivity {
    int cur[][] = {{2, 0, 0, 1}, {0, 1, 3, 0}, {0, 0, 2, 0}, {4, 3, 0, 0}, {0, 0, 0, 4}};
    int row = 5;  int col = 4;
    ImageView pics[]=new ImageView[row*col];

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_game);
        GridLayout g = (GridLayout) findViewById(R.id.griddy);
        int m=0;
        for(int i=0; i<row; i++){
            for(int j=0; j<col; j++){
                pics[m]=new ImageView(this);
                setpic(pics[m], m);
                pics[m].setId(m);
                pics[m].setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v){
                        int pos = v.getId();
                        int x = pos/col;
                        int y = pos%col;
                    }
                });
                g.addView(pics[m]);
                m++;
            }
        }
    }
    public void setpic(ImageView i, int pos){
        int x = pos/col;
        int y = pos%col;
        int picnum = cur[x][y];
        if(picnum==1)
            i.setImageResource(R.drawable.bend);
        else if(picnum==2)
            i.setImageResource(R.drawable.rend);
    }
}
```

2D Arrays

[row][col]

What colour of candy is in [2][1]?

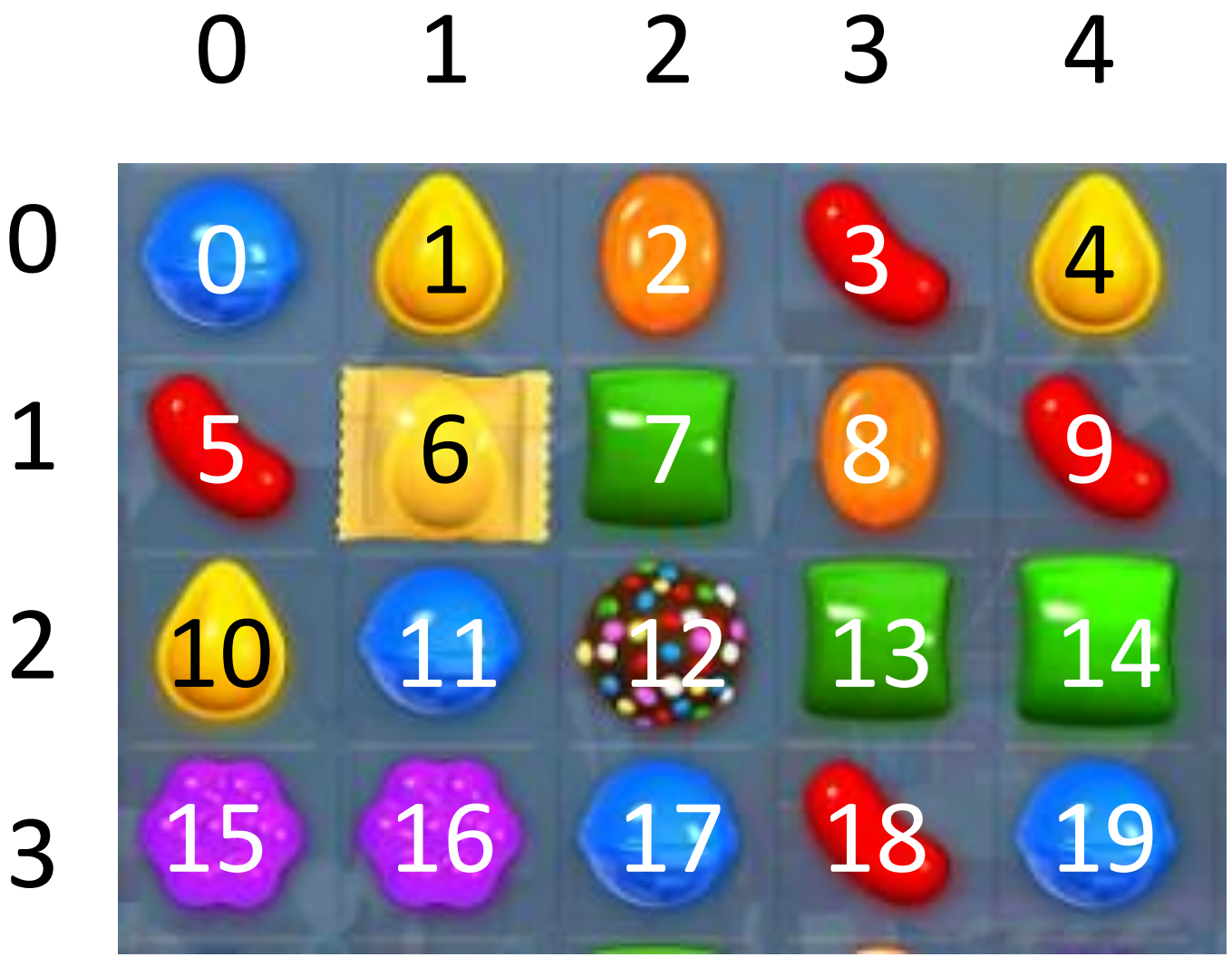


[row][col]

[2] = row,
[1] = col.
Blue.

[row][col]

What colour of candy is in [3][1]?



[row][col]

[3] = row,
[1] = col.
Purple.

```
import javax.swing.*; import java.applet.Applet; import java.awt.*; import java.awt.event.*;
public class grid extends Applet implements ActionListener
{    int row = 4;
     JButton a[] = new JButton [row * row];

     public void init ()
     {
          Panel g = new Panel (new GridLayout (row, row));
          for (int i = 0 ; i < a.length ; i++)
          {    a [i] = new JButton (" ");
               a [i].addActionListener (this);
               a [i].setActionCommand ("" + i);
               g.add (a [i]);
          }
          add (g);
          resize (180, 150);
     }

     public void actionPerformed (ActionEvent e)
     {    int n = Integer.parseInt (e.getActionCommand ());
          int x = n / row;
          int y = n % row;
          showStatus ("(" + x + ", " + y + ")");
     }
}
```

Two arrays are needed.

The image array is 1D.
It's actionCommands (or IDs in Android) will be 0-19.

The int array to track the screen will be 2D.
This will allow easy manipulation behind the scenes.

Conversion between 1D and 2D arrays is easy.

[row][col]
[2][1] = blue candy

pos / col = x
11 / 5 = 2

pos % col = y
11 % 5 = 1

[row][col]
[2][3] = green candy

pos / col = x
13 / 5 = 2

pos % col = y
13 % 5 = 3

[row][col]
[3][1] = purple candy

pos / col = x
16 / 5 = 3

pos % col = y
16 % 5 = 1

Android works the same way.

You will have a 2D tracking array AND a 1D widget array.

```
package ca.gorskicompsci.www.mediumflowfree;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.GridLayout;
import android.widget.ImageView;

public class Game extends AppCompatActivity {

    int sol [][] ={{2, 6, 6, 1}, {7, 1, 3, 8}, {7, 7, 7, 8}, {4, 3, 8, 8}, {9, 9, 9, 4}};
    int cur[][] = {{2, 0, 0, 1}, {0, 1, 3, 0}, {0, 0, 2, 0}, {4, 3, 0, 0}, {0, 0, 0, 4}};


    int row = 5;
    int col = 4;


    ImageView pics[]=new ImageView[row*col];
```

Don't mess with the top classes and items!

Set up your row and column variables.

2D int tracking answer and solution arrays

1D Widget Array

We will need to link our code to the XML.

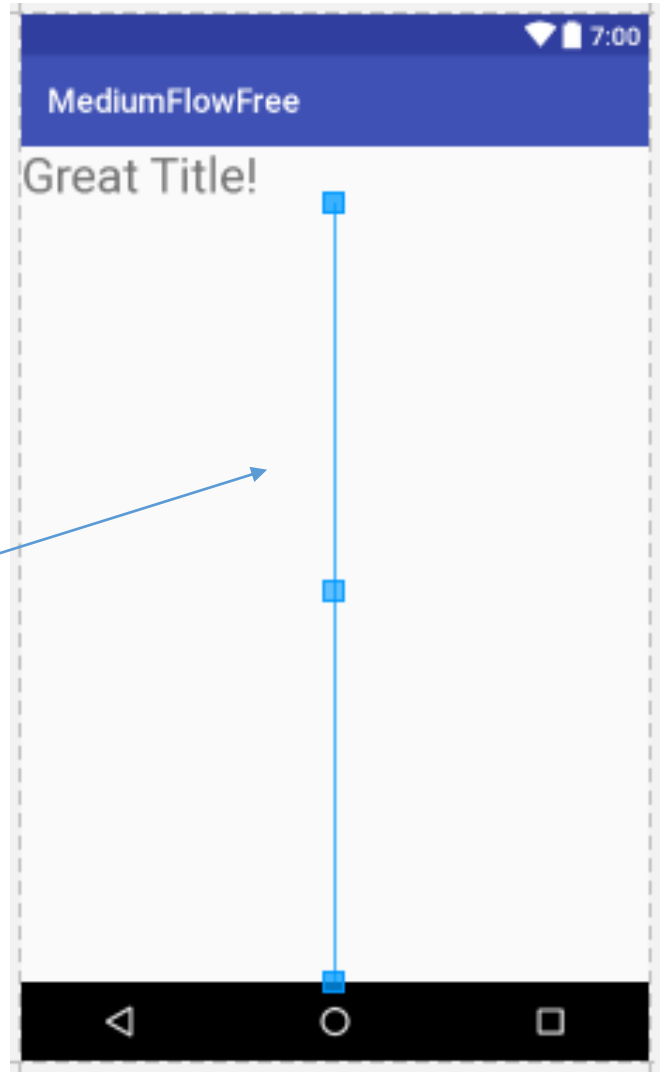
This involves making a blank grid array.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Great Title!"
        android:textSize="30sp"/>

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_gravity="center"
        android:layout_height="match_parent"
        android:rowCount="5"
        android:columnCount="4"
        android:id="@+id/griddy">
    </GridLayout>

</LinearLayout>
```



To hold the image array later on. It's empty now.

The following code makes the 1D button array.

It puts it in the Grid we created in the XML.

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

    GridLayout g = (GridLayout) findViewById(R.id.griddy);

    int m=0;
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            pics[m]=new ImageView(this);
            setpicStart(pics[m], m);
            pics[m].setId(m);
            pics[m].setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v){
                    int pos = v.getId();
                    setpic(pics[pos], pos);
                }
            });
            g.addView(pics[m]);
            m++;
        }
    }
}
```

Don’t touch. Already there.

Import your grid.

Loop through your 2D array.

Set the pictures to the starting position.

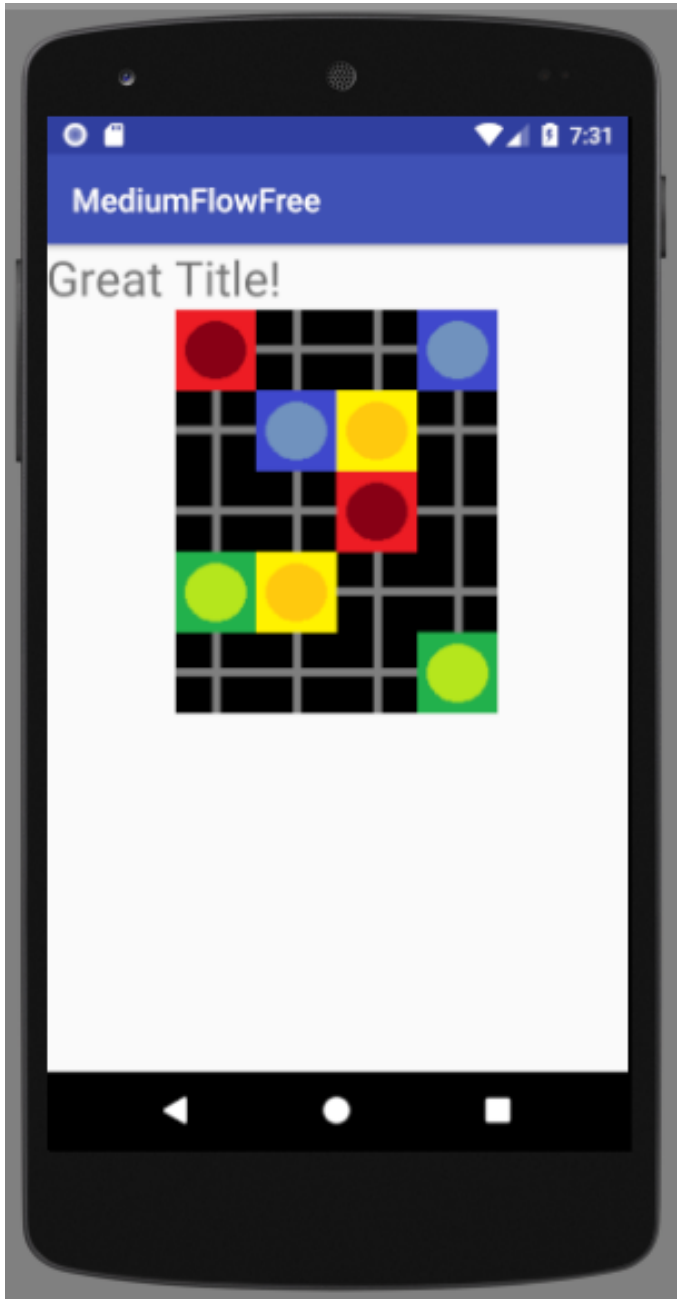
Respond to the user click.

Another problem is the images.

Assign each a number (matching the 2D tracking) and make one method & if to handle it all.

```
int cur[][] = {{2, 0, 0, 1}, {0, 1, 3, 0}, {0, 0, 2, 0}, {4, 3, 0, 0}, {0, 0, 0, 4}};
```

```
public void setpicStart(ImageView i, int pos){
    int x = pos/col;
    int y = pos%col;
    int picnum = cur[x][y];
    if(picnum==1)
        i.setImageResource(R.drawable.bend);
    else if(picnum==2)
        i.setImageResource(R.drawable.rend);
    else if(picnum==3)
        i.setImageResource(R.drawable.yend);
    else if(picnum==4)
        i.setImageResource(R.drawable.gend);
    else if(picnum==6)
        i.setImageResource(R.drawable.b);
    else if(picnum==7)
        i.setImageResource(R.drawable.r);
    else if(picnum==8)
        i.setImageResource(R.drawable.y);
    else if(picnum==9)
        i.setImageResource(R.drawable.g);
    else if(picnum==0)
        i.setImageResource(R.drawable.start);
}
```



The Complete Code Listing.

Be careful about cutting and pasting.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Great Title!"
        android:textSize="30sp"/>

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_gravity="center"
        android:layout_height="match_parent"
        android:rowCount="5"
        android:columnCount="4"
        android:id="@+id/mygrid">
    </GridLayout>

</LinearLayout>
```

```
public class Game extends AppCompatActivity {
    int cur[][] = {{2, 0, 0, 1}, {0, 1, 3, 0}, {0, 0, 2, 0}, {4, 3, 0, 0}, {0, 0, 0, 4}};
    int row = 5;  int col = 4;
    ImageView pics[]=new ImageView[row*col];

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_game);
        GridLayout g = (GridLayout) findViewById(R.id.griddy);
        int m=0;
        for(int i=0; i<row; i++){
            for(int j=0; j<col; j++){
                pics[m]=new ImageView(this);
                setpic(pics[m], m);
                pics[m].setId(m);
                pics[m].setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v){
                        int pos = v.getId();
                        int x = pos/col;
                        int y = pos%col;
                    }
                });
                g.addView(pics[m]);
                m++;
            }
        }
    }
    public void setpic(ImageView i, int pos){
        int x = pos/col;
        int y = pos%col;
        int picnum = cur[x][y];
        if(picnum==1)
            i.setImageResource(R.drawable.bend);
        else if(picnum==2)
            i.setImageResource(R.drawable.rend);
    }
}
```