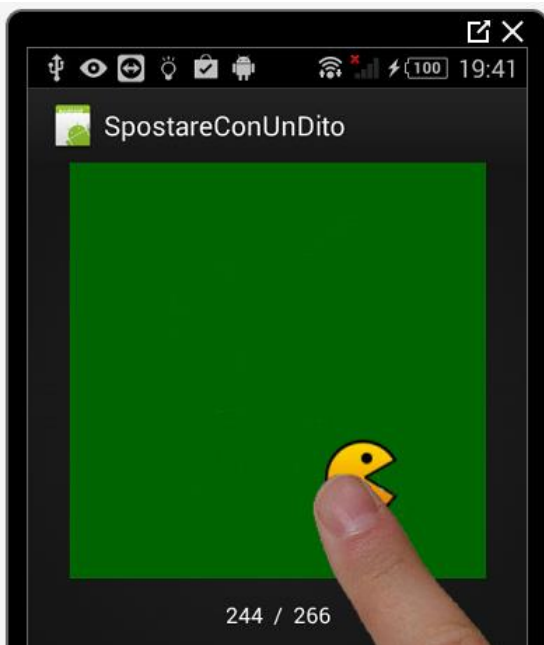


Esempio Pilota:  
**SpostareConUnDito**

**activity\_main.xml - interfaccia grafica**



`ImageView imvAreaDisegno`  
`TextView tvMessaggio`

**activity\_main.xml - codice XML**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imvAreaDisegno"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@android:drawable/ic_menu_gallery"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvMessaggio"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="....."
        app:layout_constraintTop_toBottomOf="@+id/imvAreaDisegno"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

**MainActivity.java - Codice JAVA**

```
package itis.teamsviluppatori.spostareconundito;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.graphics.Bitmap;
```

```
import android.graphics.BitmapFactory;
```

```
import android.graphics.Canvas;
```

```
import android.graphics.Color;
```

```
import android.graphics.Paint;
```

```
import android.os.Bundle;
```

```
import android.view.MotionEvent;
```

```
import android.view.View;
```

```
import android.widget.ImageView;
```

```
import android.widget.TextView;
```

```
import android.view.View.OnClickListener;
```

```
public class MainActivity extends AppCompatActivity
{
```

*/\* Per gestire un **Evento** (es. Touch) su una **View** (es. ImageView) è necessario creare un "oggetto **LISTENER**" e **associarlo** alla View.*

*Per creare l'oggetto **LISTENER** è necessario creare una classe (es. **miaClasseTouchListener**), che **IMPLEMENTI** una opportuna **INTERFACCIA** (es. **ImageView.OnTouchListener**).*

*Una **INTERFACCIA** è un po' come una **Classe Astratta**: ha dei metodi astratti che **devono essere ridefiniti** quando la si deriva.*

*Nel nostro caso, quindi, qui **definiamo una nuova classe** che "implementa" l'interfaccia "ImageView.OnTouchListener" ridefinendone il metodo "onTouch"... \*/*

```
public class miaClasseTouchListener
    implements ImageView.OnTouchListener
{
```

*/\* Il metodo **onTouch** DEVE essere implementato... (verrà eseguito quando si fa Touch sulla ImageView) \*/*

```
@Override
public boolean onTouch ( View V, MotionEvent e )
{
```

*// Riconosci l'azione controllando e.getAction() ...*

```
switch ( e.getAction() )
{
```

*// Usa l'enumerazione **MotionEvent** e i suoi valori // per capire quale azione e' specificata in e.getAction() ...*

```
case MotionEvent.ACTION_DOWN :
```

```
case MotionEvent.ACTION_MOVE :
```

```
// GetX e GetY indicano la posizione del "touch" ...
```

```
X = (int) e.getX() - ImmaginePacman.getWidth() / 2;
Y = (int) e.getY() - ImmaginePacman.getHeight() / 2;
```

```
tvMessaggio.setText( (X + " / " + Y).toString() );
```

```
Ridisegna();
break;
```

```
case MotionEvent.ACTION_UP :
```

```
tvMessaggio.setText (
    "TRASCINAMENTO TERMINATO");
```

```
break;
```

```
default: break;
```

```
}
```

```
return true;
```

```
}
```

```
}
```

```
// Dichiaro le VIEW ...
```

```
ImageView imvAreaDisegno;
```

```
TextView tvMessaggio;
```

```
// Dichiaro l'oggetto CANVAS ...
```

```
Canvas C;
```

```
// Dichiaro l'oggetto immagine che conterra' PACMAN ...
```

```
Bitmap ImmaginePacman;
```

```
// Definisci un "pennello" da utilizzare ...
```

```
Paint Rosso = new Paint();
```

```
// Variabili per mantenere la posizione del PACMAN ...
```

```
int X = 200;
```

```
int Y = 200;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState)
```

```
{
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

```
// Crea la VIEW di classe ImageView necessaria per disegnare
```

```
imvAreaDisegno = findViewById(R.id.imvAreaDisegno);
```

```
tvMessaggio = findViewById(R.id.tvMessaggio);
```

```
// Crea una immagine (AreaDisegno)
```

```
di 400x400 pixel e assegnala alla ImageView
```

```
Bitmap AreaDisegno = Bitmap.createBitmap(400, 400,
    Bitmap.Config.ARGB_8888);
```

```
imvAreaDisegno.setImageBitmap(AreaDisegno);
```

```
// Recupera la risorsa l'immagine Pacman.png e ponila
nell'oggetto ImmaginePacman
```

```
ImmaginePacman =
```

```
BitmapFactory.decodeResource(
    getResources(),
    R.drawable.pacman);
```

```
// Crea l'oggetto CANVAS sulla base dell'imm. AreaDisegno ...
```

```
C = new Canvas(AreaDisegno);
```

```
// Creiamo l'oggetto della nostra nuova classe "miaClasseListener".
```

```
// Si tratta dell'oggetto "ascoltatore" (listener) ...
```

```
miaClasseTouchListener mioListener =
    new miaClasseTouchListener();
```

```
// Impostiamolo come listener per gestire il Touch sull'immagine ...
```

```
imvAreaDisegno.setOnTouchListener (mioListener);
```

```
Ridisegna();
```

```
}
```

```
private void Ridisegna()
```

```
{
```

```
// Colora tutta l'area di verde scuro ...
```

```
C.drawColor(Color.GREEN);
```

```
// Disegna il PACMAN ...
```

```
C.drawBitmap(ImmaginePacman, X, Y, Rosso);
```

```
// Costringi Android a ritracciare imgAreaDisegno ...
```

```
imvAreaDisegno.invalidate();
```

```
}
```

```
}
```