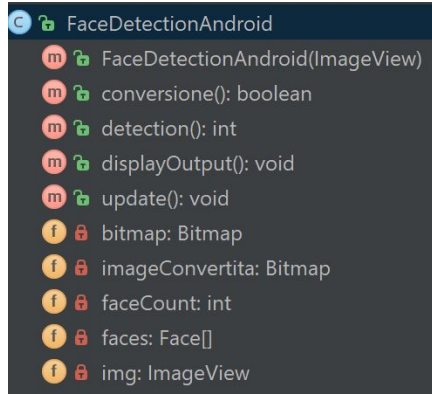


Esercizi Quarta Parte

Android MultiThread

Android face detector & AsyncTask

L'SDK standard di android contiene una libreria adibita alla face detection. La classe AndroidFaceDetection che trovate allegata a queste slides e che ha la struttura riportata in figura, usa questa libreria.



1. Il costruttore prende in ingresso un'ImageView.
2. conversione: converte la bitmap in ingresso in un formato utilizzabile dalla libreria di face detection.
3. detection: l'algoritmo di detection vero e proprio
4. displayOutput: disegna un cerchio rosso ove sono state rilevate delle facce.
5. update : aggiorna la bitmap con le facce rilevate nell'imageView dell'activity

Cliccando su un Button, l'activity deve caricare dalla cartella "res/drawable" un'immagine precedentemente salvata. Cliccando l'altro Button, si crei un AsyncTask per il calcolo delle facce, che deve quindi eseguire in un thread a parte. Alla fine di ogni fase (metodi di FaceDetectionAndroid), una ProgressBar deve aggiornarsi in maniera appropriata. Vedere la figura a lato. Bonus: se l'emulatore lo consente, fare lo stesso con una immagine presa da fotocamera.

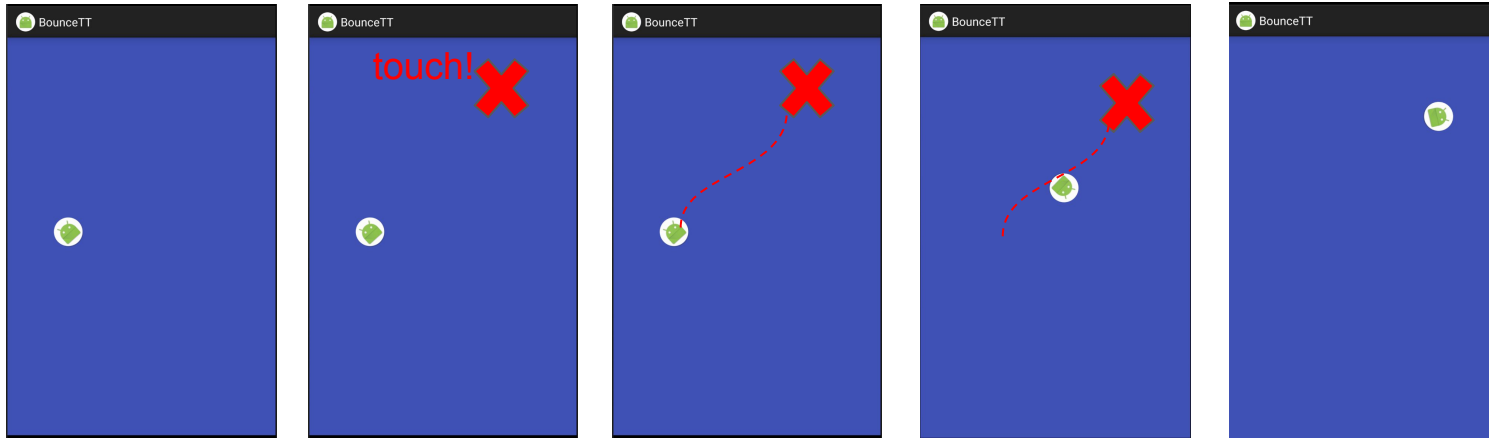
```
Bitmap imageBitmap =  
BitmapFactory.decodeResource(<activity>.getApplicationContext().getResources(),  
R.drawable.faces); //il file si chiama "faces.jpg" in "res/drawable"
```



TimerTask & Handlers

Creare una empty activity ed impostare un `AbsoluteLayout` (ignorete i warning) come unico layout di sfondo. Creare un `ImageView` in un punto arbitrario del layout di sfondo (vedere figure).

Associare un opportuno ascoltatore per i semplici eventi touch sul layout: un `TimerTask` permetterà all'`ImageView` (con una certa frequenza di aggiornamento) di spostarsi verso il punto toccato dall'utente. La comunicazione tra `TimerTask` ed il main thread deve avvenire tramite `Handler`. Gestire eventi di locking, waiting, notify e dispose del timer/timertask in maniera appropriata.



```
AbsoluteLayout.LayoutParams params = (AbsoluteLayout.LayoutParams) imageView.getLayoutParams();  
int x = params.x; int y = params.y;  
imageView.setLayoutParams(params);
```

JobService & JobInfo

Partendo da una empty activity, si crei una activity come in figura.

Le due View nella parte superiore si chiamano Switch.

I due switch permettono di descrivere le condizioni per cui schedulare un JobService.

Il controllo delle condizioni di schedulabilità avviene una volta che l'utente preme il pulsante "Schedula".

Il JobService esegue in un thread a parte e non fa altro che scrivere nel Log ciclicamente. Alla fine di ogni scrittura, il thread dorme per un periodo pre-stabilito (`Thread.sleep(...)`). Usare le funzionalità dell'emulatore per verificare il corretto funzionamento di questa app.

