



Realizzazione di un'app Android con ambiente di sviluppo integrato Android Studio

Pennisi Luigi

il 27/05/2022

5 IC

Introduzione

Ho sviluppato 2 applicazioni in android studio, la prima denominata “LuckyCoin” per esplorare l’ambiente e capirne le potenzialità e la seconda “Crypt” per fare qualcosa di sfizioso sfruttando la crittografia.

La prima simula il gioco della testa o croce, ha 2 activity: una che propone il classico gioco con la presenza anche di un Easter Egg; e l’altra che mostra un resoconto delle probabilità.

La seconda app invece sfrutta la crittografia di Vigenère per crittografare messaggi usando una chiave a scelta per poi comunicarli e decrittografarli con la stessa app con l’apposita activity.

LuckyCoin



Ecco come si presenta l'interfaccia iniziale dell'applicazione, questa è l'activity_main.xml collegata al MainActivity.java presenta 2 semplici buttoni collegati a una funzione nell'activity che apre un nuovo intent e quindi nuove activity: 1 per il classic game e un'altra per il plus.

Codice Java che apre una nuova activity con un intent:

```
public void OpenClassic(View view) {
    Intent in = new Intent(this, ClassicActivity.class);
    startActivity(in);
}
```

Così è come si presenta l'interfaccia del “Classic Game”:



Cliccando sulla moneta con un meccanismo di funzioni “onClick” l’immagine viene sostituita da una gif che simula la rotazione della moneta, allo stesso modo cliccando sulla gif viene generato un numero randomico 0 o 1 che definisce se è uscita testa o croce e in base al risultato viene sostituita la gif con l’immagine inherente.

Ho inserito anche una funzione per rendere possibile il risultato della moneta che cade dritta in piedi

```
Random random = new Random();
int res = random.nextInt(2);
if (dritto) {
    iv.setImageResource(R.drawable.euroinpiedi);
    tv2.setText("E' CADUTA DRITTA !1!1!");
    tv2.setTextColor(Color.parseColor("#B17926"));
    x = 200;
    dritto=false;
    //cD++;
} else if (res == 0) {
    iv.setImageResource(R.drawable.euro);
    tv2.setText("TESTA");
    tv2.setTextColor(Color.parseColor("#B17926"));
    //cT++;
} else {
    iv.setImageResource(R.drawable.euroback);
    tv2.setText("CROCE");
    tv2.setTextColor(Color.parseColor("#B17926"));
    //cC++;
}
```

Così è come funziona invece la “Plus Version”:



A seguito dell'input del numero di volte che si vuole lanciare la moneta viene generata una statistica che elenca il numero di volte che è uscita una certa faccia e la percentuale annessa, inutile dire che si aggira intorno al 50% anche se come possiamo vedere non è quasi mai così precisamente.

```
Random random = new Random();
for(i=0;i<x;i++){
    res = random.nextInt(2);
    if (res==0){
        cT++;
    }else{
        cC++;
    }
}
float Percent = cT*100/x;
String PT = String.valueOf(Percent);
Percent = cC*100/x;
String PC = String.valueOf(Percent);
String T = String.valueOf(cT);
String C = String.valueOf(cC);

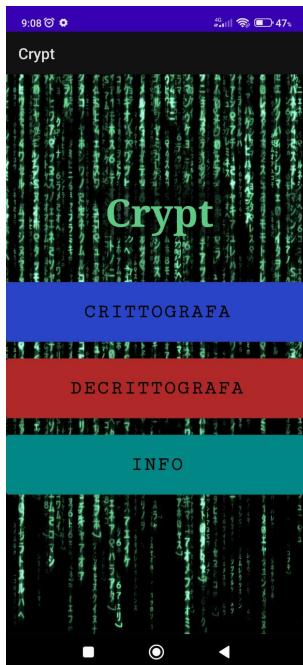
TextView tv = (TextView) findViewById(R.id.Result);
tv.setVisibility(View.VISIBLE);
tv.setText("Resoconto:\n\nTeste: "+T+" ("+PT+"% )\nCroci:t: "+C+" ("+PC+"% )");
```

[Clicca qui](#) per visualizzare il codice ClassicActivity.java

[Clicca qui](#) per visualizzare il codice PlusActivity.java

Crypt

Interfaccia principale da cui possiamo scegliere quale operazione fare:



Da qui cliccando su Info ci escono le istruzioni per chi non se ne intende cybersecurity:



Interfaccia “Crittografa”:



Nell’immagine a sinistra, ho fatto un esempio di crittografia con la chiave “chiavesegretissima”, il testo che vediamo nel box bianco invece è la frase generata dalla crittografia eseguita con il tasto esegui.



Quando si critta e decrypta nella chiave non ci vanno messi ne numeri ne spazi ne altri caratteri strani, in qualunque caso l’app limita automaticamente la digitazione.

Interfaccia “Decrittografa”:



Evvtz fyums amhmdptvn, rgr ummmdg m
ohhzjrw

Questa è pressoché identica a quella di prima cambia ovviamente solo il nome in alto e la funzione nel backend se così si può definire il file activity.java che sta dietro l'activity.xml.

Dopo aver inserito la stessa chiave usata per criptare e nel box bianco il testo criptato cliccando su esegui partirà la funzione “decripta” che modificare il contenuto del box bianco EditText nel risultato della decriptazione



Se volete sapere il messaggio che ho crittografato non vi resta che installare l'apk e decriptarlo con la chiave: “chiavesegretissima”.

Mex. "Evvtz fyums amhmdptvn, rgr ummmdg m ohhzjrw".

I codici per criptare e decriptare sono lunghi se volete dargli un’occhiata qui sotto ci sono i link alle intere activity.java con tutte le funzioni usate.

[Clicca qui](#) per il visualizzare il codice di CryptActivity.java

[Clicca qui](#) per il visualizzare il codice di DecryptActivity.java

I file dei main non li allego dato che contengono solo funzioni che sul click aprono avviano nuovi intent.

The Fine