Esercitazione

Prima parte - Creare l'activity di lancio

1) Creare un progetto Android di nome "IndovinaIlNumero" e dare nome alla activity di default "StartActivity"

2) Creare un nuovo layout di nome "start.xml" cho contenga in un LinearLayout:
    i) una TextView con label "Nome Utente Proprio" ed una corrispondente EditText
    ii) una seconda TextView con label "Nome Utente Avversario" ed un'altra EditText
    iii) un bottone con label "Gioca"

3) Impostare il layout creato nella StartActivity

4) Creare una seconda activity con nome "Main" ed impostare il layout main.xml come root element

5) Nel metodo onCreate della StartActivity, gestire l'evento del bottone lanciando un Intent per lanciare la activity "Main" impostando come
paramtetri esterni della activity i valori delle due EditText

6) Nella activity main visualizzare i valori delle due edit taxt nella textView

7) Eseguire e testare il programma

Seconda parte - Creare la connessione ed inizializzare il gioco

1) Scaricare il file xmpp.zip al link http://ge.tt/9w3TJ1E scompattare ed aggiungere i file al progetto

2) Far estendere all'activity Main l'interfaccia MessageReceiver ed implementare il metodo richiesto

3) Inizializzare la connessione usando la classe fornita

```
    connection = new ConnectionManager(nomeProprio, nomeAvversario, this);
```

4) Definire nella classe un enum che contenga gli stati dell'applicazione:

```
    enum Stato {
        WAIT_FOR_START,WAIT_FOR_START_ACK
    }
```

5) definire un timer ed una azione associata la timer

```
    Timer timer = new Timer();
    TimerTask sendStart = new TimerTask() {
        @Override
        public void run() {
            // TODO Auto-generated method stub
            if (statoCorrente == Stato.WAIT_FOR_START_ACK) {
                connection.send("START");
            } else {
                Log.d(TAG, "Sending START but the state is " + statoCorrente);
            }
        }
    };
```

6) nel onCreate decidere chi comincia mediante le hash dei nomi utenti: chi cominica invia il messaggio di start periodicamente usando il timer

```
        if(nomeAvversario.hashCode()<nomeProprio.hashCode()){
         // Inizio io
         timer.schedule(sendStart, 1000, 5000);
         statoCorrente=Stato.WAIT_FOR_START_ACK;
        } else{
         // Inizia lui
         //Io aspetto il pacchetto;
         statoCorrente=Stato.WAIT_FOR_START;
        }
```

7) Creare un handler che riceva il messaggio SHOW_TOAST e mostri un toast con la string contenuta con chiave "toast"

```
    final Handler handler = new Handler() {
         @Override
         public void handleMessage(android.os.Message msg) {
              switch (msg.what) {
              case Main.SHOW_TOAST:
                   Toast.makeText(Main.this,msg.getData().getString("toast"),
Toast.LENGTH_LONG).show();
                   break;
              default:
                   super.handleMessage(msg);
              }
         }
    };
```

8) gestire i messaggi ricevuti nel metodo receiveMessage

i) se ricevo lo start invio l'ack ed informo l'utente che dee scegliere un numero

```
         if (body.equals("START")) {
              if (statoCorrente == Stato.WAIT_FOR_START) {
                   // Mando l'ack indietro
                   connection.send("STARTACK");
                   Message osmsg = handler.obtainMessage(Main.SHOW_TOAST);
                   Bundle b = new Bundle();
                   b.putString("toast", "Scegli un numero");
                   osmsg.setData(b);
                   handler.sendMessage(osmsg);
                   statoCorrente=Stato.USER_SELECTING;
              } else {
                   Log.e(TAG, "Ricevuto START ma lo stato è " + statoCorrente);
              }
         }
```

ii) allungando l'if precedente, ricevo lo start_ack e aspetto l'avversario che gioca:

```
          else if (body.equals("STARTACK")) {
              if (statoCorrente == Stato.WAIT_FOR_START_ACK) {
                   statoCorrente=Stato.WAIT_FOR_NUMBER_SELECTION;
              } else {
                   Log.e(TAG, "Ricevuto STARTACK ma lo stato è " + statoCorrente);
              }
         }
```

9) aggungere nel layout tre bottoni con label "1", "2" e "3" e gestire i tre click con il metodo numberSelected che invia all'avvarsario con il messaggio "SELCTED:X" con il numero scelto al posto della X

```
    public void numberSelected(View v){
        Button b = (Button) v;
        b.getText().toString();
        if (statoCorrente == Stato.USER_SELECTING) {
            connection.send("SELECTED:"+b.getText().toString());
            statoCorrente=Stato.WAIT_FOR_BET;
        }
    }
```

10) gestire nel metodo receiveMessage il messaggio "SELECTED"

```
        else if (body.startsWith("SELECTED")) {
            if (statoCorrente == Stato.WAIT_FOR_NUMBER_SELECTION){
                selectedNumber = body.split(":")[1];
                Message osmsg = handler.obtainMessage(Main.SHOW_TOAST);
                Bundle b = new Bundle();
                b.putString("toast", "Indovina il numero");
                osmsg.setData(b);
                handler.sendMessage(osmsg);
                statoCorrente=Stato.USER_BETTING;
            }else{
                Log.e(TAG, "Ricevuto SELECTED ma lo stato è " + statoCorrente);
            }

        }
```

11) gestire la scelta dell'utente che deve indovinare e mostrare il risultato

```
    else if (statoCorrente == Stato.USER_BETTING) {
            String bet = b.getText().toString();
            connection.send("BET:"+bet);
            if(bet.equals(selectedNumber)){
                Toast.makeText(Main.this,"Bravo hai indovinato, ora tocca te",
Toast.LENGTH_LONG).show();
            } else{
                Toast.makeText(Main.this,"Peccato non hai indovinato, ora tocca te",
Toast.LENGTH_LONG).show();
            }
            statoCorrente=Stato.USER_SELECTING;
        }
```

12) gestire nel metodo receiveMessage il messaggio "BET" e mostare il risultato all'utente

```
else if (body.startsWith("BET")) {
            if (statoCorrente == Stato.WAIT_FOR_BET){
                String result  = body.split(":")[1];
                Message osmsg = handler.obtainMessage(Main.SHOW_TOAST);
                Bundle b = new Bundle();
                if(result.equals("Y"))
                    b.putString("toast", "Hai perso, il tuo avversario ha
indovinato");
                else
                    b.putString("toast", "Hai vinto, il tuo avversario ha sbagliato");
                osmsg.setData(b);
                handler.sendMessage(osmsg);
                statoCorrente=Stato.WAIT_FOR_NUMBER_SELECTION;
            }else{
                Log.e(TAG, "Ricevuto SELECTED ma lo stato è " + statoCorrente);
            }

        }
```