Guess the number

Software Documentation

Author: matiwa

Table of contents

Table of contents……………………………………………………………………………….2

Introduction………………………..……………………………………………………...3

Describing of the application’s operation……………………………………………………...3

What is needed for use?..13

Algorithm used……………………………………………………………………………..13

Interface description………………………………………………………………………..14

Source code description……………………………………………………………………15

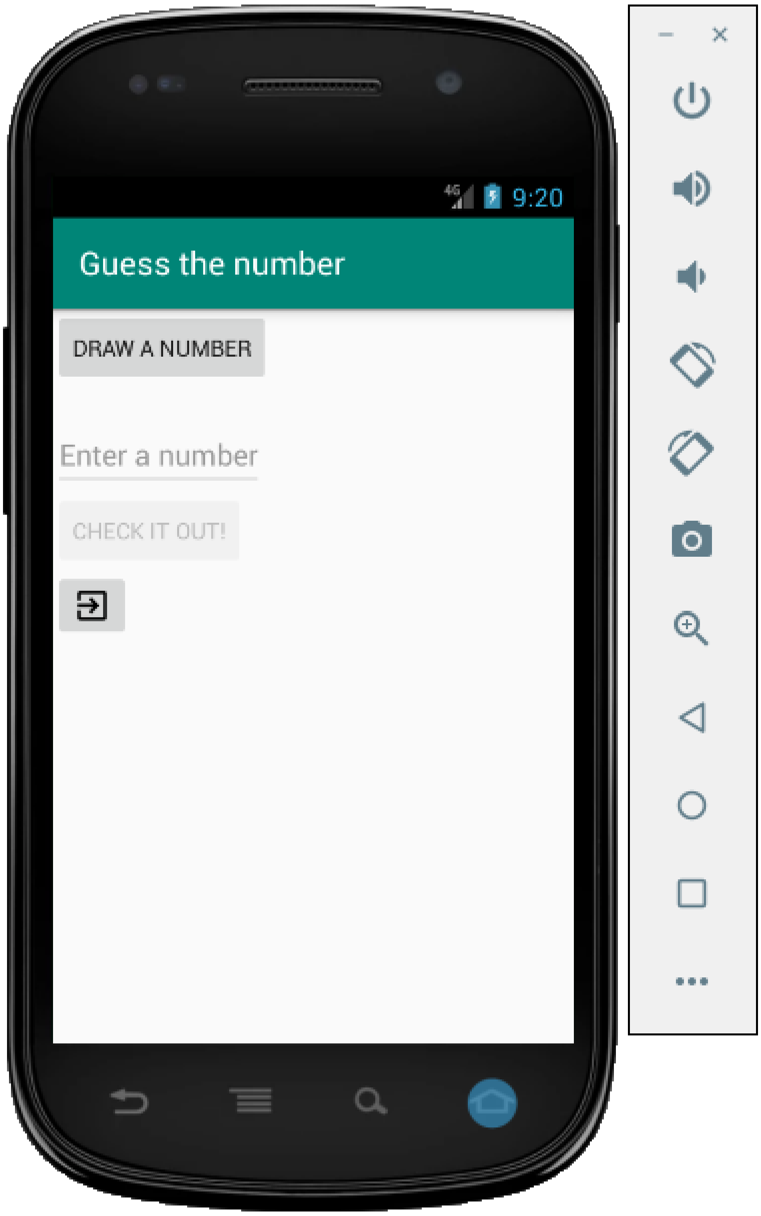
List of drawings……………………………………………………………………………19

List of listings………………………………………………………………………………19

Introduction

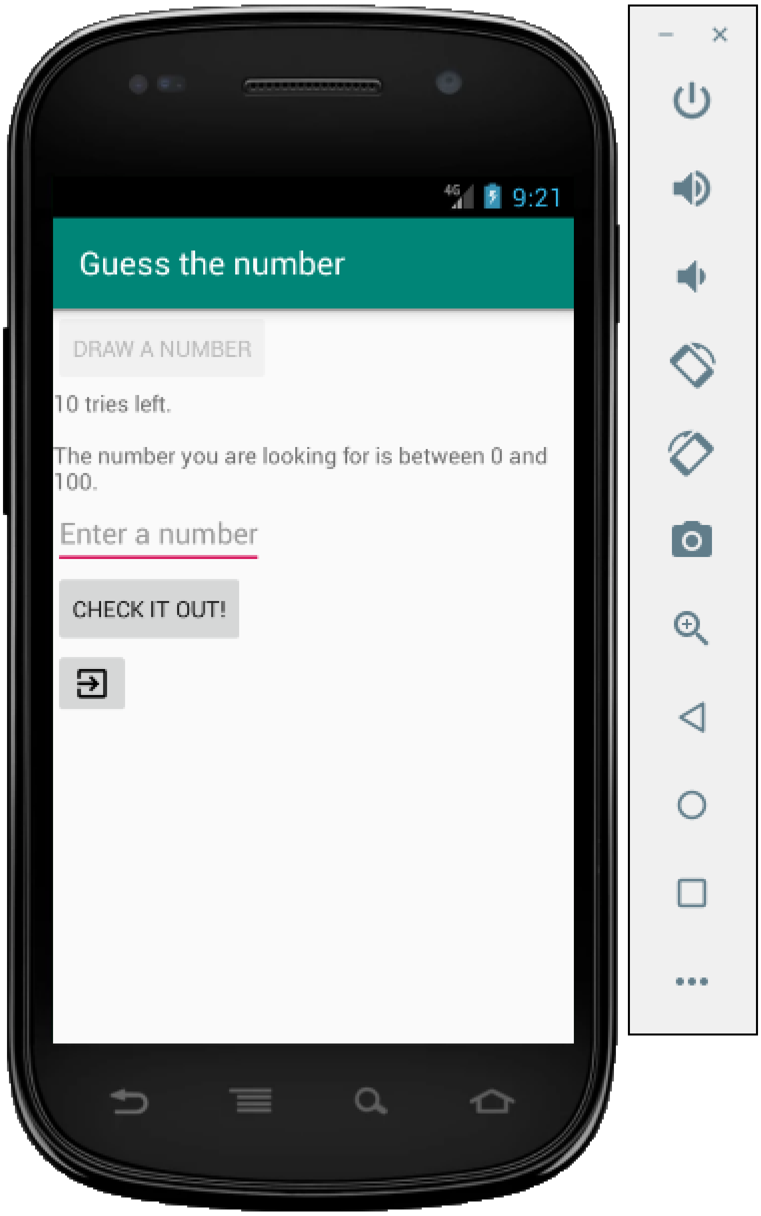
This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application is used to guess the drawn number.

Describing of the application's operation



Drawing 1: The beginning of the application's operation [own study]

The user needs the "Draw Number" button for the program to draw the number. After this action, an appropriate message will appear in the text box that a number from 0 to 100 has been drawn, with the fact that there are 10 attempts to guess the number. The complete test sequence is shown below. There are messages in the tooltip with hints as to the drawn number, if the entered number is too small or too large. There are also messages informing about the number of remaining tries that are real-time in the text box. A balloon message about guessed number and won appears. A lost balloon message also appears with information about the number the user was looking for. The characteristics of the application and the accompanying messages characterize a typical game of chance. After guessing (winning) or losing, the user can play again. The loss message was presented without errors.



9:21

Guess the number

DRAW A NUMBER

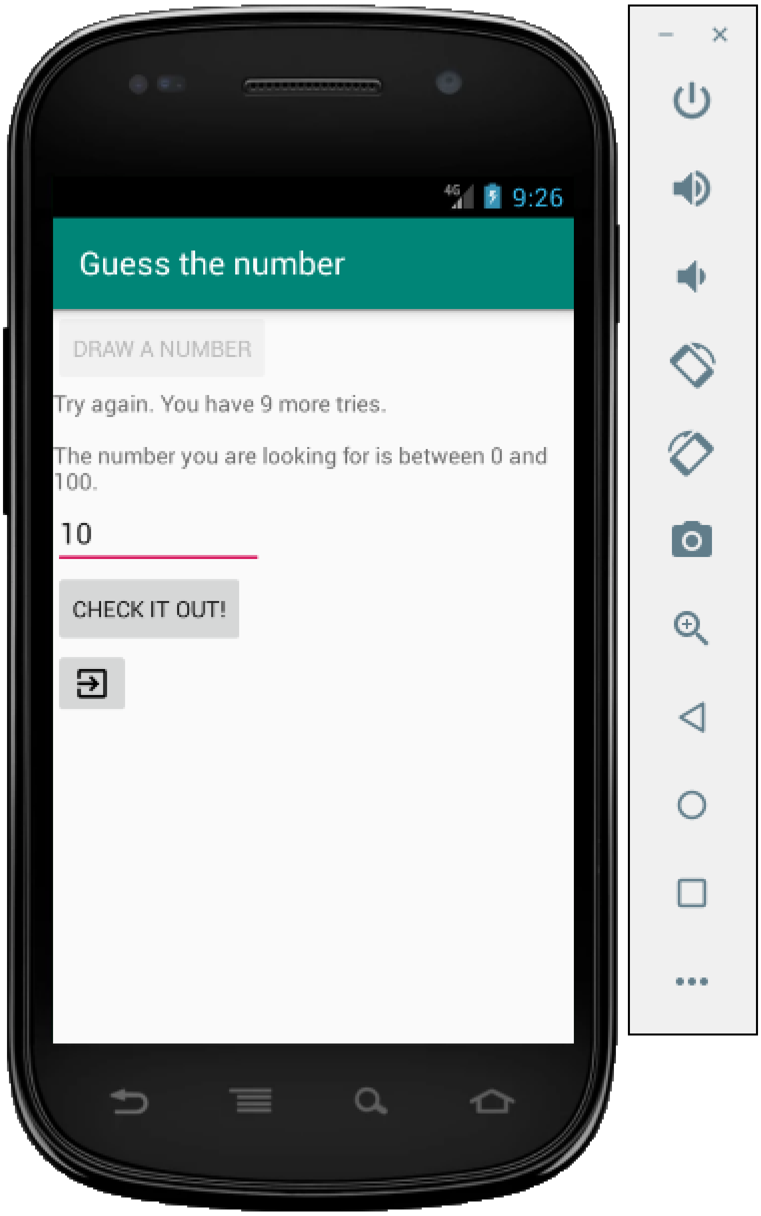
10 tries left.

The number you are looking for is between 0 and 100.

Enter a number

CHECK IT OUT!

Drawing 2: First step [own study]



9:26

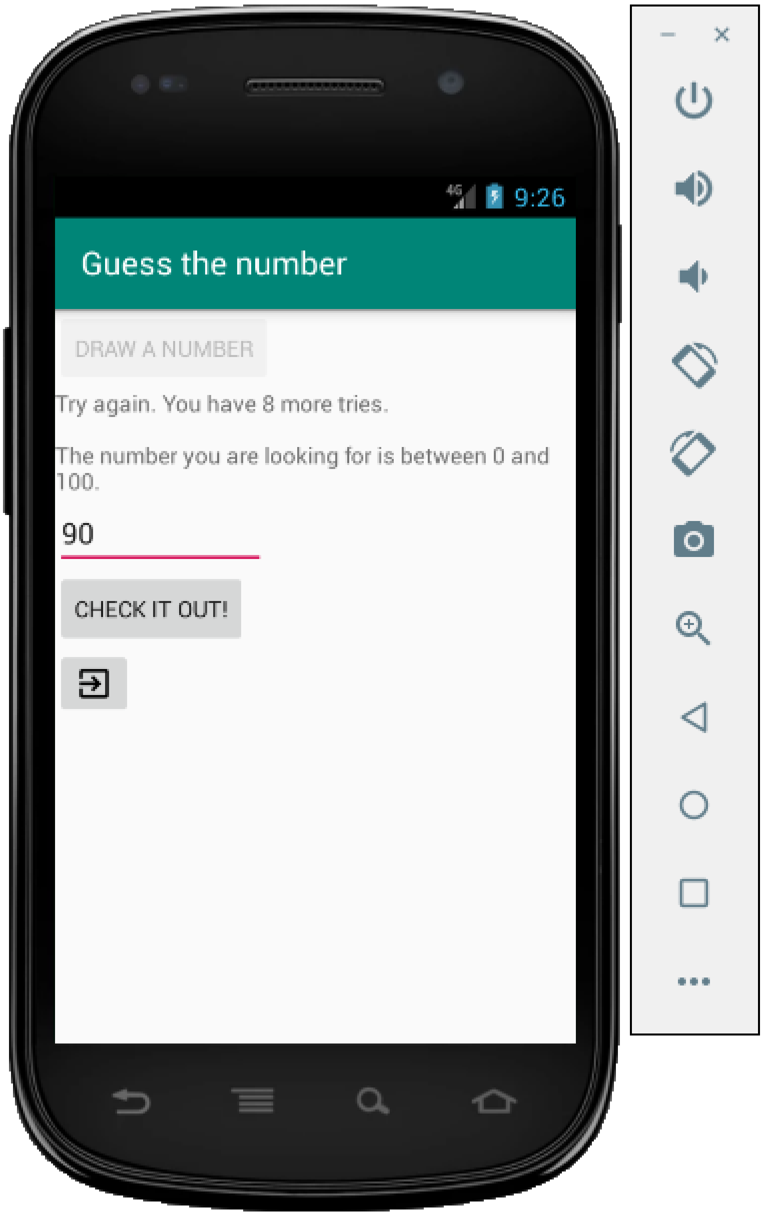
Guess the number

Try again. You have 9 more tries.

The number you are looking for is between 0 and 100.

10

Drawing 3: Second step [own study]



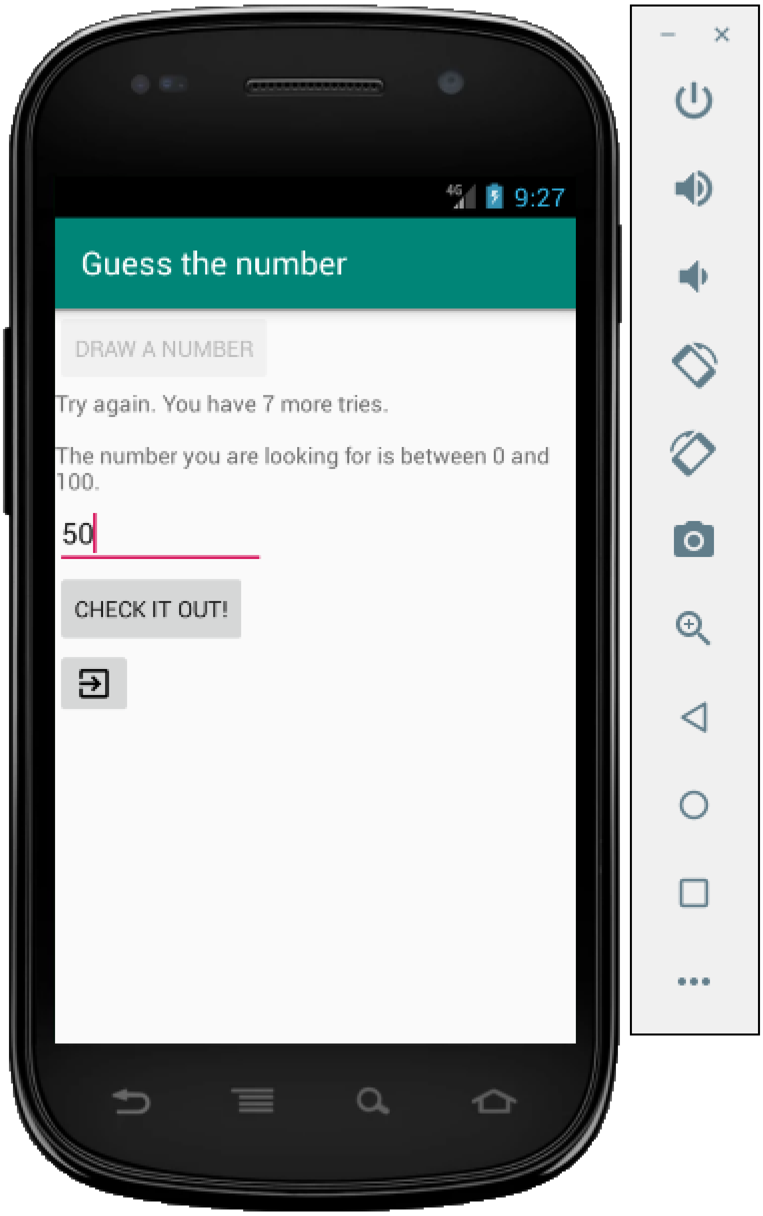
Guess the number

Try again. You have 8 more tries.

The number you are looking for is between 0 and 100.

90

Drawing 4: Third step [own study]

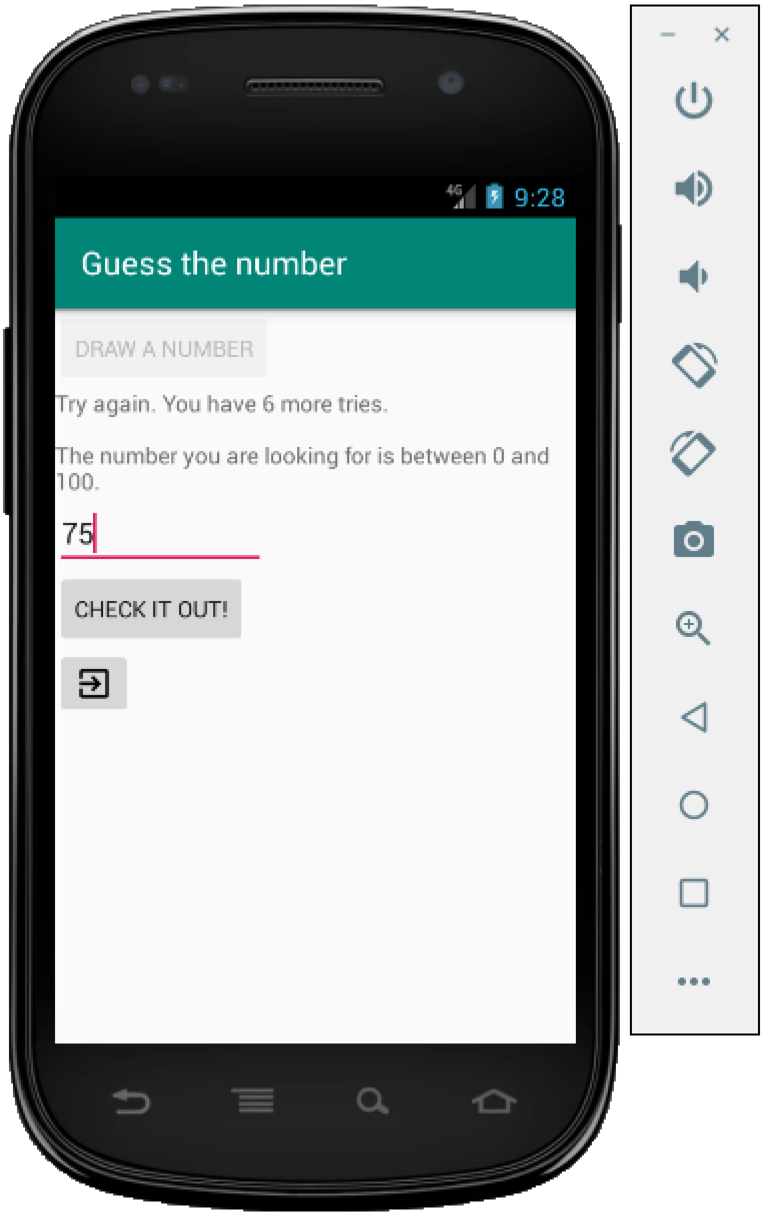


Try again. You have 7 more tries.

The number you are looking for is between 0 and 100.

50

Drawing 5: Fourth step [own study]



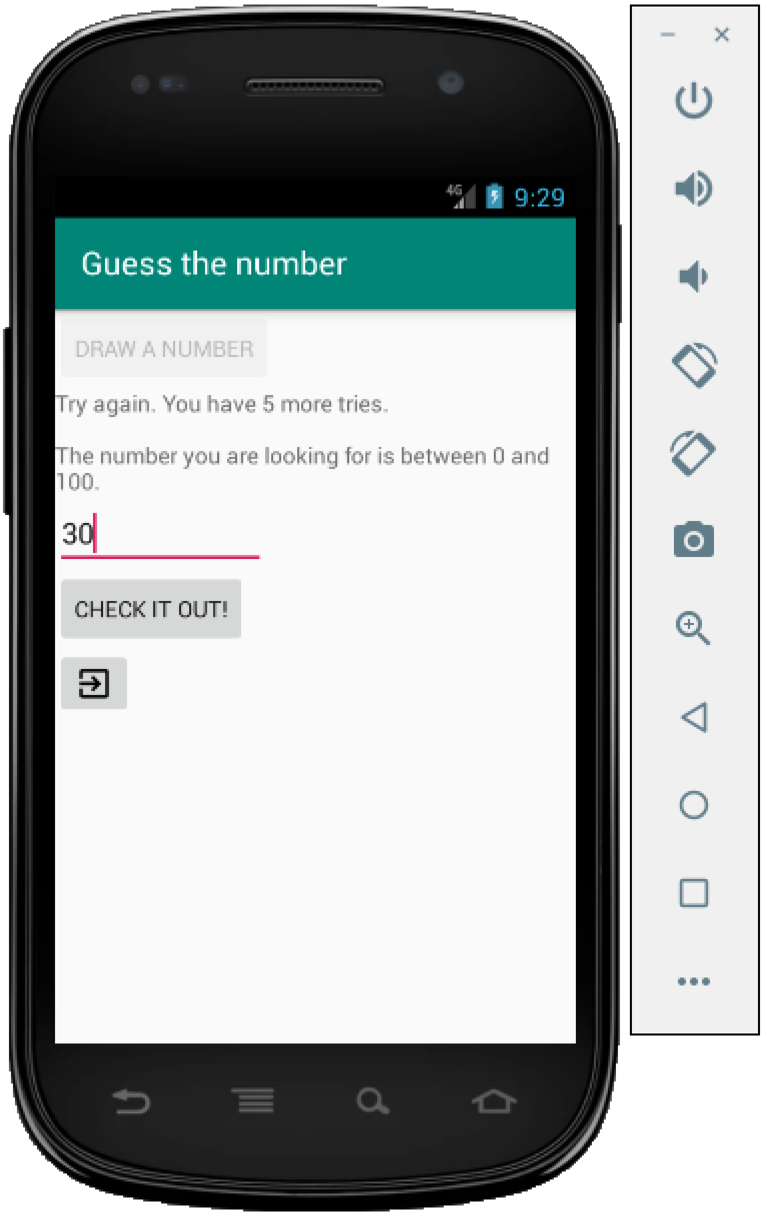
Guess the number

Try again. You have 6 more tries.

The number you are looking for is between 0 and 100.

75

Drawing 6: Fifth step [own study]



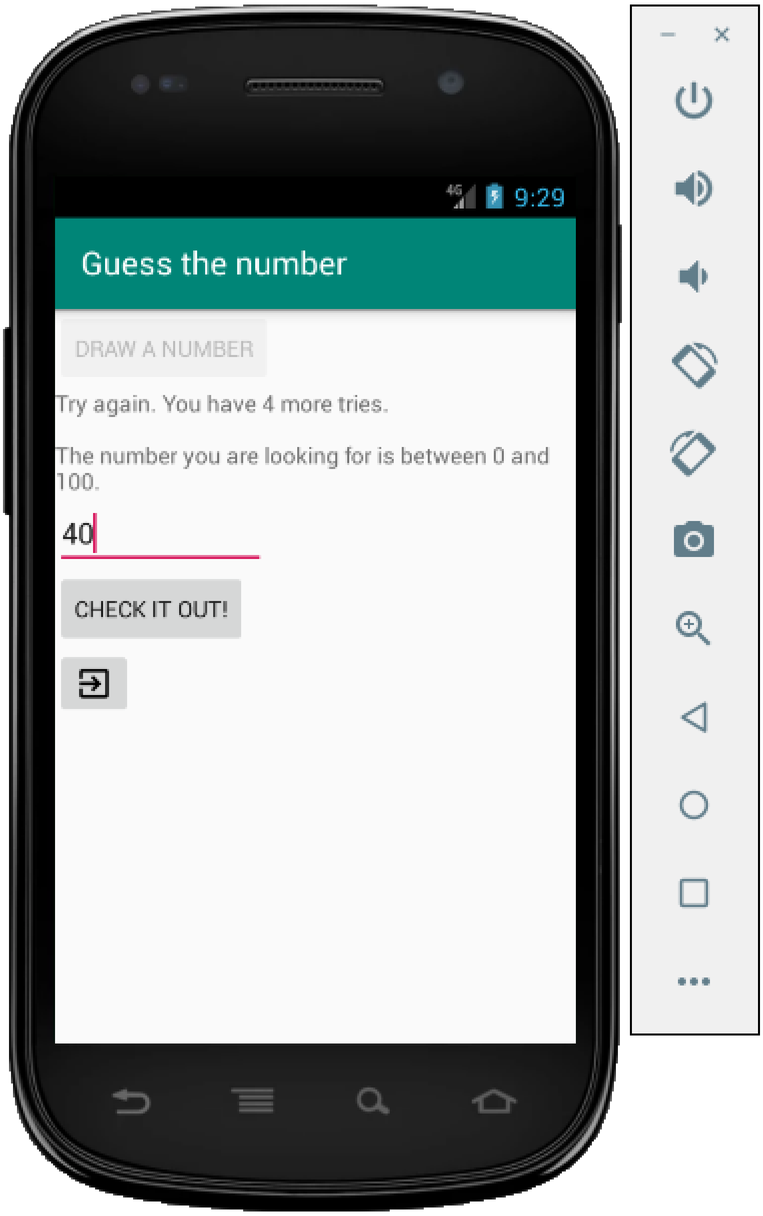
Guess the number

Try again. You have 5 more tries.

The number you are looking for is between 0 and 100.

30

Drawing 7: Sixth step [own study]



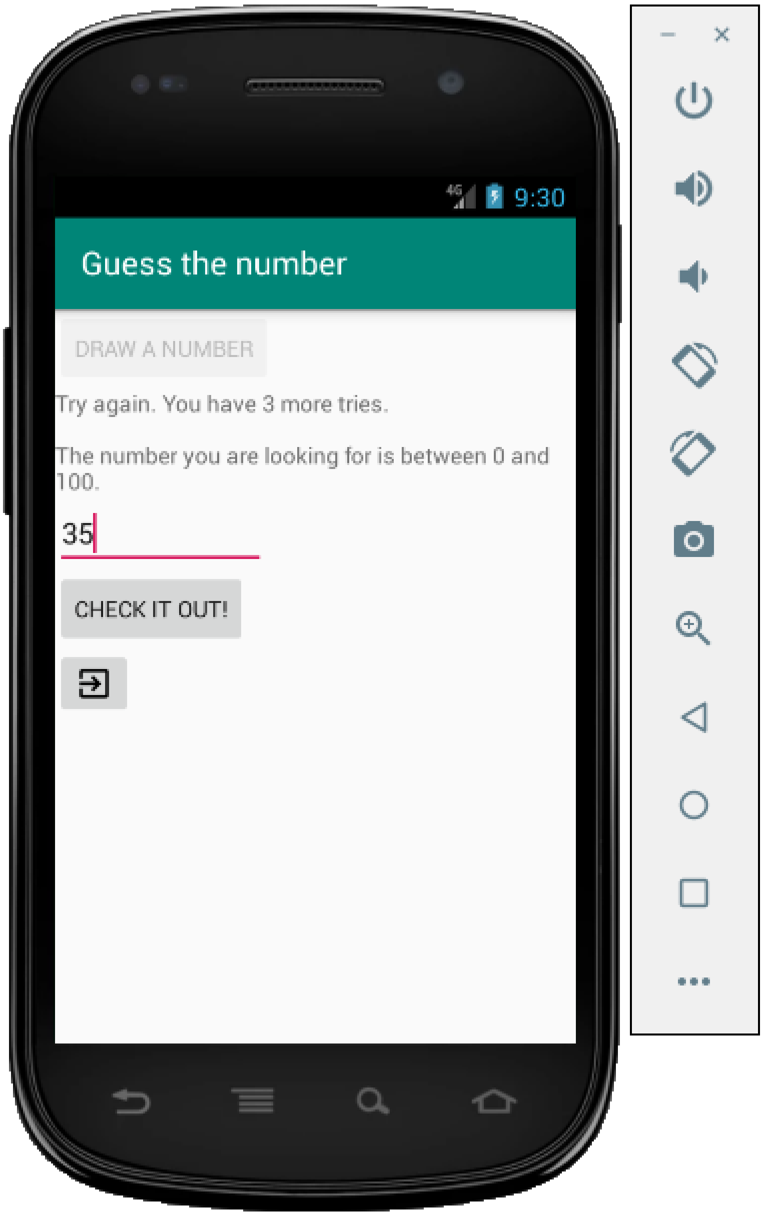
Guess the number

Try again. You have 4 more tries.

The number you are looking for is between 0 and 100.

40

Drawing 8: Seventh step [own study]



9:30

Guess the number

DRAW A NUMBER

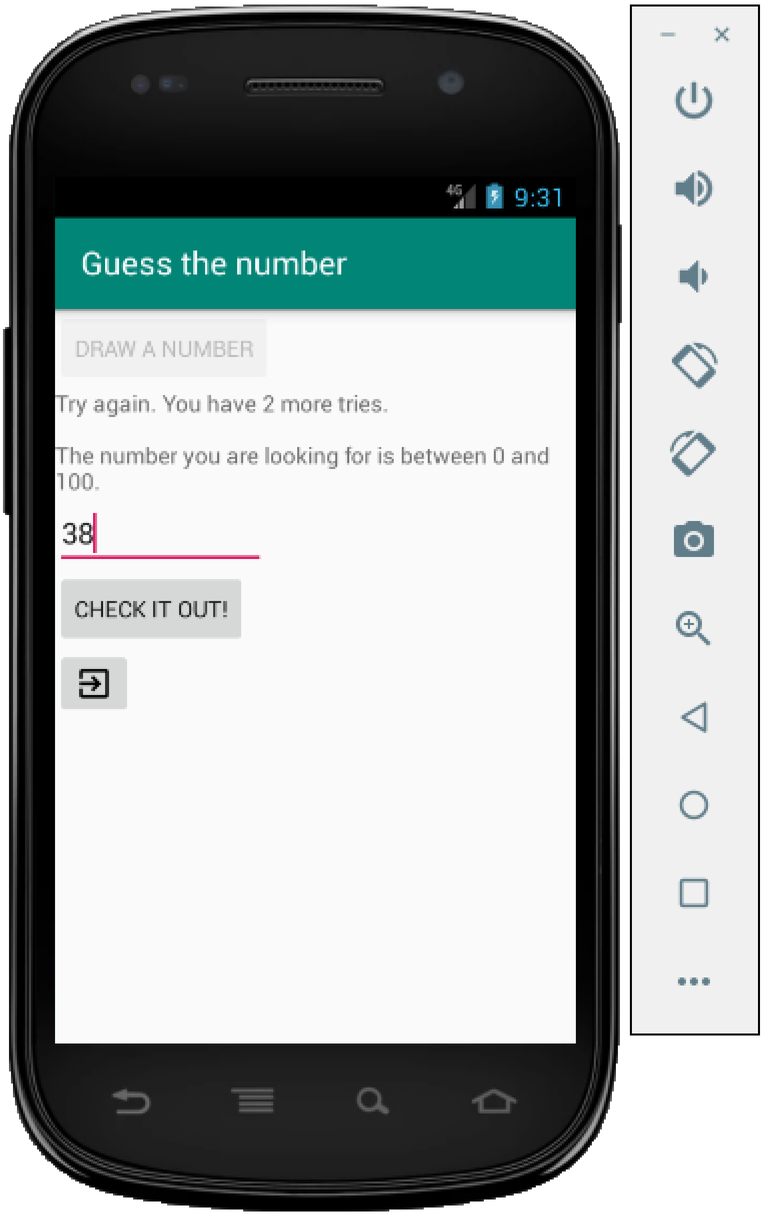
Try again. You have 3 more tries.

The number you are looking for is between 0 and 100.

35

CHECK IT OUT!

Drawing 9: Eighth step [own study]



Guess the number

DRAW A NUMBER

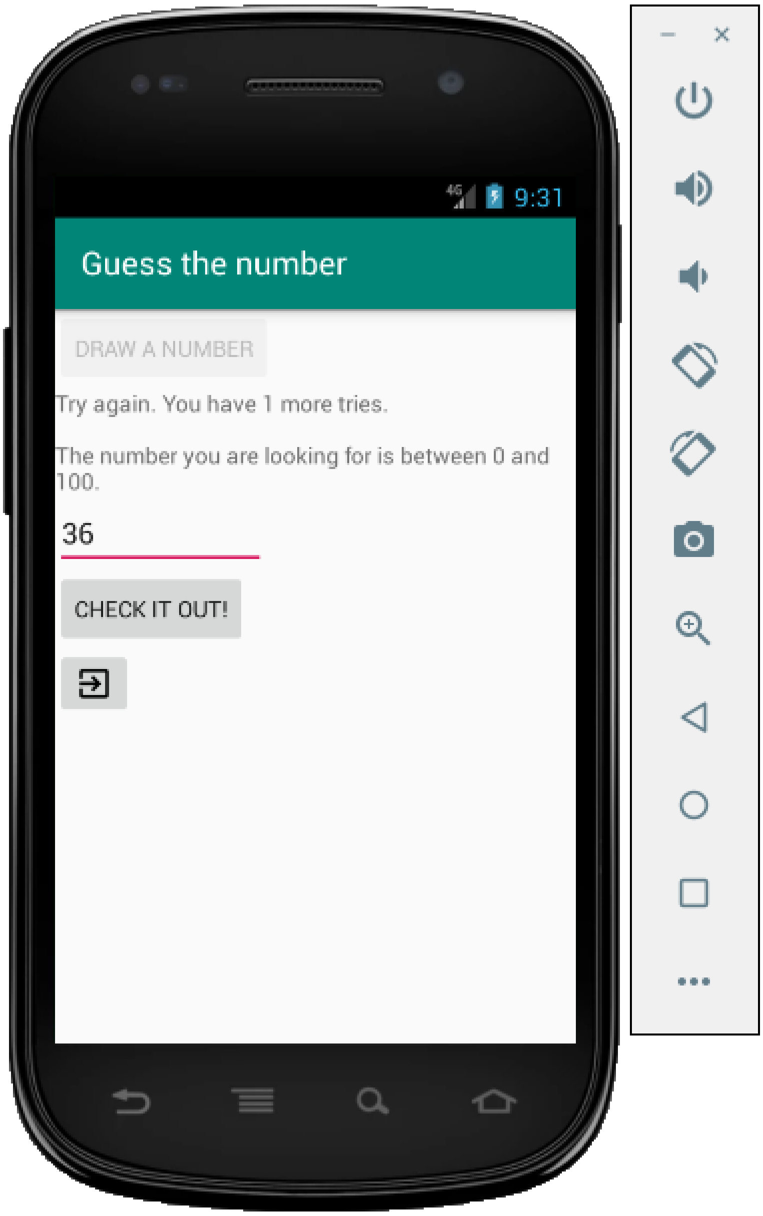
Try again. You have 2 more tries.

The number you are looking for is between 0 and 100.

38

CHECK IT OUT!

Drawing 10: Ninth step [own study]



Try again. You have 1 more tries.

The number you are looking for is between 0 and 100.

36

Drawing 11: Tenth step

This is the correct use of the program. There may be coding errors that have not been detected by the developer.

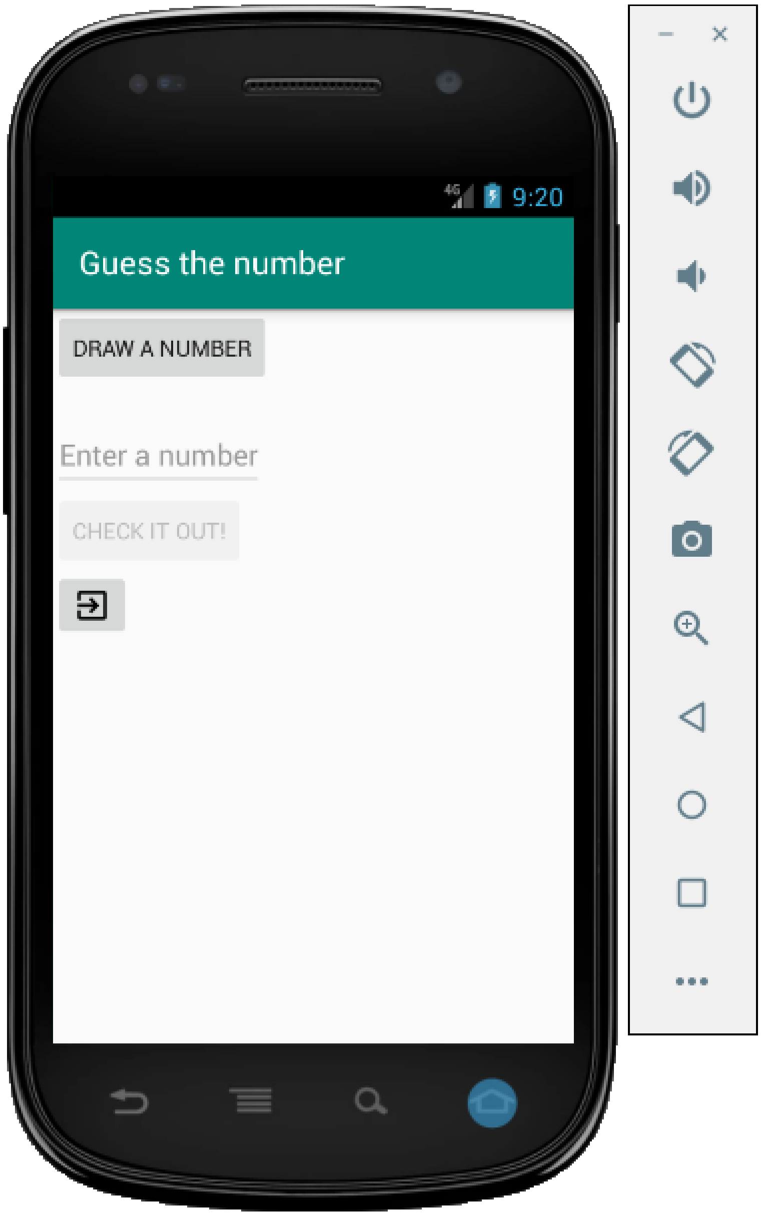
What is needed for use?

The application requires installation on the Android operating system with a minimum API level Android 4.0 (IceCreamSandwich), i.e. on all mobile devices with this system.

Algorithm used

The basic form of the algorithm can be deduced from the previous section. To sum up, the application is a typical game of guessing the drawn number. The user has 10 tries to guess it. He has hints on what to follow when guessing when he makes a mistake. He is kept informed about the number of attempts he has left. If he wins, he has information about it. If he loses, he knows it too. After one of these two messages, the game starts all over again.

Interface description



Drawing 12: Graphical interface [own study]

The interface has the basic components available in the Android Studio development environment: TextView, ImageButton, Button and EditText.

Source code description

The project was made in the Java programming language, in the Android Studio programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
package com.example.guessthenumber;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Button;
import android.widget.Toast;

import java.util.Random;

public class MainActivity extends AppCompatActivity {
    EditText eean;
    TextView tvnot;
    Button btndan,btnchio;
    ImageButton imge;
    private int not;//number of tries
    private int nd;//number drawn
    private int ne;//number entered
    private Random r;

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

        initializecomponents();
        buttonsoperation();
    }

    private void initializecomponents(){
        eean=findViewById(R.id.eean); eean.setEnabled(false);
        tvnot=findViewById(R.id.tvnot); tvnot.setEnabled(false);
        btndan=findViewById(R.id.btndan);
        btnchio=findViewById(R.id.btnchio);
btnchio.setEnabled(false);
        imge=findViewById(R.id.imge);
    }
    private void buttonsoperation(){
        btndan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                btnchio.setEnabled(true);
                eean.setEnabled(true);
                tvnot.setEnabled(true);
                r=new Random();
                nd=r.nextInt(101);
                not=10;
                tvnot.setText(""+not+" tries left.\r\n\r\nThe number
you are looking for is between 0 and 100.");
                btndan.setEnabled(false);
            }
        });
        btnchio.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try{
                    ne=Integer.parseInt(eean.getText().toString());
                    if(ne!=nd && not>0){
                        if(ne<nd){

Toast.makeText(getApplicationContext(),"The number is too
small",Toast.LENGTH_LONG).show();
                        }else if(ne>nd){

Toast.makeText(getApplicationContext(),"The number is too
big",Toast.LENGTH_LONG).show();
                        }
                        not--;
                        tvnot.setText("Try again. You have "+not+"
more tries.\r\n\r\nThe number you are looking for is between 0 and
100.");
                    }else if(ne==nd){
                        btndan.setEnabled(true);
                        btnchio.setEnabled(false);
                        tvnot.setEnabled(false); tvnot.setText("");
                        eean.setEnabled(false); eean.setText("");

Toast.makeText(getApplicationContext(),"Bravo!!! The drawn number is
actually "+ne,
                                Toast.LENGTH_LONG).show();
                    }
                    if(not<1){
                        btndan.setEnabled(true);
                        btnchio.setEnabled(false);
                        tvnot.setEnabled(false); tvnot.setText("");
                        eean.setEnabled(false); eean.setText("");
                        Toast.makeText(getApplicationContext(),"You
were looking for a number "+ne,
                                Toast.LENGTH_LONG).show();
                    }
                }catch (Exception ex) {

Toast.makeText(getApplicationContext(),ex.getMessage(),Toast.LENGTH_
LONG).show();
                }
            }
        });
        imge.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
    }
}
```

Listing 1: The source code for MainActivity.java file [own study]

```
<?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">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <Button
            android:id="@+id/btndan"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Draw a number" />

        <TextView
            android:id="@+id/tvnot"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="" />
        <EditText
            android:id="@+id/eean"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:hint="Enter a number"
            android:inputType="number"
            android:numeric="integer" />

        <Button
            android:id="@+id/btnchio"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Check it out!" />

        <ImageButton
            android:id="@+id/imge"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/ic_exit_to_app_black_24dp" />
    </LinearLayout>

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

Listing 2: The source code for the activity_main.xml file [own study]

List of drawings

Drawing 1: The beginning of the application's operation [own study]………………………...3

Drawing 2: First step [own study]……………...……………………………………………4

Drawing 3: Second step [own study]……………………..………………………………5

Drawing 4: Third step [own study]……………………………………..………………...6

Drawing 5: Fourth step [own study]……………………….………………………….…7

Drawing 6: Fifth step [own study]……………………………………..……………...8

Drawing 7: Sixth step [own study]……………………….…………………………...9

Drawing 8: Seventh step [own study]……………………..10

Drawing 9: Eighth step [own study]……………………….…………………………...11

Drawing 10: Ninth step [own study]…………………………………………………….12

Drawing 11: Tenth step [own study]……………………………..……………………...13

Drawing 12: Graphical interface [own study]…………………………………………..…14

List of listings

Listing 1: The source code for MainActivity.java [own study]..……………………………..15
Listing 2: The source code for activity_main.xml file[own study]…………….…………..17