

```

package com.example.adriana.kitty;

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.net.Uri;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

import com.google.android.gms.appindexing.Action;
import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.common.api.GoogleApiClient;

public class MyActivity extends AppCompatActivity {

    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        /** Se enlazan los elementos declarados en el layout al código */
        _Boton_Arriba_Izquierda = (Button)
        findViewById(R.id.activity_my_boton_arriba_izquierda);
        _Boton_Arriba_Medio = (Button)
        findViewById(R.id.activity_my_boton_arriba_medio);
        _Boton_Arriba_Derecha = (Button)
        findViewById(R.id.activity_my_boton_arriba_derecha);

        /**Agregue los demás botones*/

        _Titulo_Turno_Jugador = (TextView)
        findViewById(R.id.activity_my_titulo_turno_jugador);

        _Tablero = new int[3][3];
        /** Se inicializa el turno al jugador 1 */
        _Turno = false;
        cambiarTurno();
        _TotalMovimientos = 0;
        _FinPartida = false;

        /** Se asocian los listeners a los botones */
        _Boton_Arriba_Izquierda.setOnClickListener(new
        OnBotonClickListener(_Boton_Arriba_Izquierda, 0, 0));
        _Boton_Arriba_Medio.setOnClickListener(new
        OnBotonClickListener(_Boton_Arriba_Medio, 0, 1));
    }
}

```

```

        _Boton_Arriba_Derecha.setOnClickListener(new
OnBotonClickListener(_Boton_Arriba_Derecha, 0, 2));

        _Boton_Medio_Izquierda.setOnClickListener(new
OnBotonClickListener(_Boton_Medio_Izquierda, 1, 0));
        _Boton_Medio_Medio.setOnClickListener(new
OnBotonClickListener(_Boton_Medio_Medio, 1, 1));
        _Boton_Medio_Derecha.setOnClickListener(new
OnBotonClickListener(_Boton_Medio_Derecha, 1, 2));

/*Agregue los listeners de los demás botones*/

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    }

    private void cambiarTurno() {
        _Turno = !_Turno;
        _Titulo_Turno_Jugador.setText(getResources().getString(_Turno ?
R.string.jugador1 : R.string.jugador2));
    }

    private void validarJuego() {
        boolean _HileraCompleta = false;
        for (int i = 1; i < 3; i++) {

            //Filas
            if (_Tablero[0][0] == i && _Tablero[0][1] == i && _Tablero[0][2] == i) {
                _Boton_Arriba_Izquierda.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);
                _Boton_Arriba_Medio.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);
                _Boton_Arriba_Derecha.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);

                _HileraCompleta = true;
                break;
            } else if (_Tablero[1][0] == i && _Tablero[1][1] == i && _Tablero[1][2] ==
i) {
                _Boton_Medio_Izquierda.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);
                _Boton_Medio_Medio.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);
                _Boton_Medio_Derecha.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);

                _HileraCompleta = true;
                break;
            } else if (_Tablero[2][0] == i && _Tablero[2][1] == i && _Tablero[2][2] ==
i) {
                _Boton_Abajo_Izquierda.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);
                _Boton_Abajo_Medio.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);
                _Boton_Abajo_Derecha.setBackgroundResource(i == CASILLA_JUGADOR_1 ?
R.mipmap.circulo_gano : R.mipmap.x_gano);

                _HileraCompleta = true;
                break;
            }
        }
    }

```

```

//Columnas
/*Agregue el código para validar las columnas*/

//Diagonales
/*Agregue el código para validar las diagonales*/

    if (_HileraCompleta) {
        _FinPartida = _HileraCompleta;
        showDialog(DIALOG_CONFIRMACION);
    }
}

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case DIALOG_CONFIRMACION:
            AlertDialog.Builder builder3 = new AlertDialog.Builder(this);
            builder3.setMessage(R.string.reiniciar)
                .setNegativeButton(getString(R.string.NO),
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog,
                            int id) {
                            dismissDialog(DIALOG_CONFIRMACION);
                        }
                    })
                .setPositiveButton(getString(R.string.SI),
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog,
                            int id) {
                            dismissDialog(DIALOG_CONFIRMACION);
                            onCreate(null);
                        }
                    })
                .return builder3.create();
            default:
                return null;
    }
}

@Override
public void onStart() {
    super.onStart();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client.connect();
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "My Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's
content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app deep link URI is correct.
        Uri.parse("android-app://com.example.adriana.kitty/http/host/path")
    );
    AppIndex.AppIndexApi.start(client, viewAction);
}

@Override
public void onStop() {
    super.onStop();
}

```

```

// ATTENTION: This was auto-generated to implement the App Indexing API.
// See https://g.co/AppIndexing/AndroidStudio for more information.
Action viewAction = Action.newAction(
    Action.TYPE_VIEW, // TODO: choose an action type.
    "My Page", // TODO: Define a title for the content shown.
    // TODO: If you have web page content that matches this app activity's
content,

    // make sure this auto-generated web page URL is correct.
    // Otherwise, set the URL to null.
    Uri.parse("http://host/path"),
    // TODO: Make sure this auto-generated app deep link URI is correct.
    Uri.parse("android-app://com.example.adriana.kitty/http/host/path")
);
AppIndex.AppIndexApi.end(client, viewAction);
client.disconnect();
}

/**
 * Custom Listeners
 */
class OnBotonClickListener implements OnClickListener {

    @Override
    public void onClick(View v) {
        if (!_FinPartida && _TotalMovimientos <= CANTIDAD_MAXIMA_JUGADAS &&
        _Tablero[_Fila][_Columna] == 0) {
            _Tablero[_Fila][_Columna] = _Turno ? CASILLA_JUGADOR_1 :
CASILLA_JUGADOR_2;
            _Boton.setBackgroundResource(_Turno ? R.mipmap.circulo : R.mipmap.x);
            _TotalMovimientos++;

            if (_TotalMovimientos >= CANTIDAD_MINIMA_JUGADAS) {
                validarJuego();
            }

            cambiarTurno();
        } else if (_FinPartida || _TotalMovimientos <= CANTIDAD_MAXIMA_JUGADAS) {
            showDialog(DIALOG_CONFIRMACION);
        }
    }

    public OnBotonClickListener(Button pBoton, int pFila, int pColumna) {
        _Boton = pBoton;
        _Fila = pFila;
        _Columna = pColumna;
    }

    private Button _Boton;
    private int _Fila;
    private int _Columna;
}

/**
 * Atributos
 */

/**
 * Matriz para manejar la lógica de la representación del tablero
 */
private int[][] _Tablero;

```

```

/**
 * Indica el turno del jugador: True=> Jugador, False=>Jugador 2
 */
private boolean _Turno;

/**
 * Constantes de juego
 */
private int _TotalMovimientos;
private boolean _FinPartida;

private final int CANTIDAD_MINIMA_JUGADAS = 5;
private final int CANTIDAD_MAXIMA_JUGADAS = 9;
private final int CASILLA_JUGADOR_1 = 1;
private final int CASILLA_JUGADOR_2 = 2;

private static final int DIALOG_CONFIRMACION = 0;

/**
 * Elementos de UI
 */
private Button _Boton_Arriba_Izquierda;
private Button _Boton_Arriba_Medio;
private Button _Boton_Arriba_Derecha;

private Button _Boton_Medio_Izquierda;
private Button _Boton_Medio_Medio;
private Button _Boton_Medio_Derecha;

private Button _Boton_Abajo_Izquierda;
private Button _Boton_Abajo_Medio;
private Button _Boton_Abajo_Derecha;

private TextView _Titulo_Turno_Jugador;

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_my, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
}

```