

Práctica 01

Programación para dispositivos II

Primera parte (30 puntos)

Diseño de la interfaz, tomando en cuenta dos idiomas y otro estilo de aplicación.

Segunda parte (30 puntos)

Hacer funcionar la parte lógica del programa.

Tercera parte (40 puntos)

Corregir la inteligencia artificial, de tal forma que no se puede ganar en el modo “difícil”

Estilo de la aplicación.

```
<style name="AppTheme" parent="android:Theme.Holo.NoActionBar.Fullscreen">
```

Orientación de la pantalla

```
<activity
    android:screenOrientation="landscape"
</activity>
```



* La práctica se evaluará en clases.

activityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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:id="@+id/linearLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:gravity="center_horizontal|center"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.276"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.501">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TRES EN RAYA" />

    <Button
        android:id="@+id/boton1Jugador"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1 Jugador"
        android:onClick="aJugar"/>

    <RadioGroup
        android:id="@+id/configDificultad"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <RadioButton
            android:id="@+id/botonEasy"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Facil" />

        <RadioButton
            android:id="@+id/botonNormal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:checked="true"
            android:text="Normal" />

        <RadioButton
            android:id="@+id/botonInsane"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Dificil" />
```

```
</RadioGroup>

<Button
    android:id="@+id/boton2Jugadores"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="2 Jugadores"
    android:onClick="aJugar"/>
</LinearLayout>

<GridLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="80dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="80dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:columnCount="3"
    android:rowCount="3"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.802"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <ImageView
        android:id="@+id/cuadro1"
        android:src="@drawable/casilla"
        android:onClick="toque"/>

    <ImageView
        android:id="@+id/cuadro2"
        android:src="@drawable/casilla"
        android:onClick="toque"/>

    <ImageView
        android:id="@+id/cuadro3"
        android:src="@drawable/casilla"
        android:onClick="toque"/>

    <ImageView
        android:id="@+id/cuadro4"
        android:src="@drawable/casilla"
        android:onClick="toque"/>

    <ImageView
        android:id="@+id/cuadro5"
        android:src="@drawable/casilla"
        android:onClick="toque"/>

    <ImageView
        android:id="@+id/cuadro6"
        android:src="@drawable/casilla"
        android:onClick="toque"/>

    <ImageView
```

```

        android:id="@+id/cuadro7"
        android:src="@drawable/casilla"
        android:onClick="toque"/>

        <ImageView
            android:id="@+id/cuadro8"
            android:src="@drawable/casilla"
            android:onClick="toque"/>

        <ImageView
            android:id="@+id/cuadro9"
            android:src="@drawable/casilla"
            android:onClick="toque"/>
    </GridLayout>
</android.support.constraint.ConstraintLayout>

```

MainActivity.java

```

package com.example.tp_link.tresenrayafinal;

import android.app.Activity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RadioGroup;
import android.widget.Toast;

public class MainActivity extends Activity
{
    private int [] casillas;
    private int jugadores;
    private Partida partida;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        casillas=new int[9];
        casillas[0]=R.id.cuadro1;
        casillas[1]=R.id.cuadro2;
        casillas[2]=R.id.cuadro3;
        casillas[3]=R.id.cuadro4;
        casillas[4]=R.id.cuadro5;
        casillas[5]=R.id.cuadro6;
        casillas[6]=R.id.cuadro7;
        casillas[7]=R.id.cuadro8;
        casillas[8]=R.id.cuadro9;
    }
    public void aJugar(View vista)
    {
        /*Todas las casillas en blanco*/
        ImageView imagen;
        for(int cadaCasilla:casillas)
        {

```

```

        imagen=findViewById((cadaCasilla));
        imagen.setImageResource(R.drawable.casilla);
    }
    /*for(int  cadaCasilla=0;cadaCasilla<9;cadaCasilla++)
    {
        imagen=findViewById(casillas[cadaCasilla]);
        imagen.setImageResource(R.drawable.casilla);
    }*/

    jugadores=1;
    /*Verificamos si se presiono 1 jugador o 2 jugadores*/
    if(vista.getId()==R.id.boton2Jugadores)
    {
        jugadores=2;
    }

    /*Verificamos que dificultad fue seleccionada*/
    RadioGroup configDificultad=findViewById(R.id.configDificultad);
    int id=configDificultad.getCheckedRadioButtonId();
    int dificultad=0;
    if(id==R.id.botonNormal)
    {
        dificultad=1;
    }
    else
    {
        if(id==R.id.botonInsane)
        {
            dificultad=2;
        }
    }

    partida=new Partida(dificultad);/*instanseamos la clase partida*/

    (findViewById(R.id.boton1Jugador)).setEnabled(false);/*desabilita el
    boton 1 jugador*/
    (findViewById(R.id.configDificultad)).setAlpha(0);/*vuelve invisible a
    los radio group*/
    (findViewById(R.id.boton2Jugadores)).setEnabled(false);
}
public void toque(View vista)
{
    /*Si no se selecciono 1 jugador o 2 jugadores el metodo no hace nada*/
    if(partida==null)
    {
        return;
    }

    int casilla=0;/*variable para almacenar el numero de casilla que se
    toco*/
    for(int i=0;i<9;i++)
    {
        if(casillas[i]==vista.getId())/*obtenemos que vista casilla fue
        activada*/
        {
            casilla=i;
            break;/*rompe la secuencia del for*/
        }
    }

```

```

    }

    /*si es true se puede usar la casilla,
    si es false termina el metodo por esa casilla esta en uso*/

    if(partida.comprueba_casilla(casilla)==false)
    {
        return;
    }

    marcar(casilla); /*Cambia la imagen dependiendo que jugador sea*/
    int resultado=partida.turno(); //0,1,2,3
    if(resultado>0)
    {
        termina(resultado);
        return;
    }
    /*Para jugar contra la maquina*/
    if(jugadores==1)
    {
        casilla=partida.ia();
        while(partida.comprueba_casilla(casilla)==false)
        {
            casilla=partida.ia();
        }
        marcar(casilla);
        resultado=partida.turno();
        if(resultado>0)
        {
            termina(resultado);
        }
    }
}

/*Genera un mensaje dependiendo el resultado del juego:
* 1 gano circulo
* 2 gano aspa
* 3 empate
* al final resetea el juego para volver a jugar*/
private void termina(int resultado)
{
    String mensaje;
    if(resultado==1)
    {
        mensaje="Ganan los circulos";
    }
    else
    {
        if(resultado==2)
        {
            mensaje="Ganan las aspas";
        }
        else
        {
            mensaje="Empate!!!!";
        }
    }
    Toast toast=Toast.makeText(this,mensaje, Toast.LENGTH_SHORT);
    toast.setGravity(Gravity.CENTER,0,0);
}

```

```

        toast.show();

        /*Comenzar otra vez*/
        partida = null;

        findViewById(R.id.boton1Jugador).setEnabled(true);
        findViewById(R.id.configDificultad).setAlpha(1);
        findViewById(R.id.boton2Jugadores).setEnabled(true);
    }
    /*Cambia la imagen a circulo o aspa dependiendo el turno del jugador
    * que seleccione una casilla*/
    private void marcar(int casilla)
    {
        ImageView imagen=(ImageView) findViewById(casillas[casilla]);

        if(partida.jugador==1)
        {
            imagen.setImageResource(R.drawable.circulo);
        }
        else
        {
            imagen.setImageResource(R.drawable.aspa);
        }
    }
}

```

Clase Partida

```

package com.example.tp_link.tresenrayafinal;

import java.util.Random;

public class Partida
{
    public int dificultad;
    public int jugador;
    public int [] casillas;
    public final int [][]
combinaciones={{0,1,2},{3,4,5},{6,7,8},{0,3,6},{1,4,7},{2,5,8},{0,4,8},{2,4,6}
};
    public Partida(int dificultad)
    {
        this.dificultad=dificultad;
        jugador=1;
        casillas=new int [9];/*arreglo donde se inicializa todos sus campos
con valores ceros*/
        for(int i=0; i<9;i++)
        {
            casillas[i]=0;
        }
    }
    /*Retorna la casilla para poder realizar tres en raya*/
    /*retorna valores de 0 - 8 y -1 en caso que no tenga la posibilidad de
hacer 3 en raya*/
    public int dosEnRaya(int jugador_en_turno)/*jugador que esta solicitando
ver si hay 2 en raya*/
    {

```

```
int casilla=-1;//posicion q no existe en el tablero
int cuantas_lleva=0; //Si el jugador esta en 2EnRaya
for(int i=0;i<combinaciones.length;i++)
{
    for (int pos:combinaciones[i])
    {
        if(casillas[pos]==jugador_en_turno)
        {
            cuantas_lleva++;
        }
        if(casillas[pos]==0)
        {
            casilla=pos; /*Si es cero significa que se puede hacer una
jugada para ganar*/
        }
    }
    if(cuantas_lleva==2 && casilla != -1)
    {
        return casilla;
    }
    casilla=-1;
    cuantas_lleva=0;
}
return -1;
}

public int ia()
{
    int casilla; /*Variable para retornar el valor seleccionado*/

    casilla=dosEnRaya(2);

    if(casilla!=-1)
    {
        return casilla;
    }

    if(dificultad>0)
    {
        casilla=dosEnRaya(1); /*Bloquear al jugador 1 para q no gane*/
        if(casilla!=-1)
        {
            return casilla;
        }
    }

    if(dificultad==2)
    {
        if(casillas[4]==0)
        {
            return 4;
        }
        if(casillas[0]!=0 && casillas[8]!=0 && casillas[4]==2)
        {
            return 1;
        }
        if(casillas[0]==0)
        {
            return 0;
        }
    }
}
```



```

        }
        if(casillas[2]==0)
        {
            return 2;
        }
        if(casillas[6]==0)
        {
            return 6;
        }
        if(casillas[8]==0)
        {
            return 8;
        }
    }

    Random casilla_azar=new Random();
    casilla=casilla_azar.nextInt(9);
    return casilla;
}

public int turno()
{
    boolean empate=true; /*Suponemos que esta en empate hasta comprobar lo
contrario*/
    boolean ult_movimiento=true; /*true significa que el jugador gano la
partida*/
    for(int i=0;i<combinaciones.length;i++)
    {
        for (int pos:combinaciones[i])
        {
            //System.out.println("valor posicion "+i+ " "+casillas[pos]);
            if(casillas[pos]!=jugador)
            {
                ult_movimiento=false;
            }
            if(casillas[pos]==0)
            {
                empate=false; //si hay una casilla libre
            }
        }
        if(ult_movimiento)
        {
            return jugador;
        }
        ult_movimiento=true;
    }

    if(empate)
    {
        return 3; //Resulta empate
    }

    jugador++;
    if(jugador>2)
    {
        jugador=1;
    }
    return 0;
}

/*Casilla q nos retorna true si la casilla especificada esta en blanco

```

```
* false si la casilla esta marcada con circulo o aspa*/
public boolean comprueba_casilla(int casilla)
{
    if(casillas[casilla]!=0)
    {
        return false;//no se puede marcar
    }
    else
    {
        casillas[casilla]=jugador;
    }
    return true; //se puede marcar
}
}
```