TRES EN RAYA

Curso: Desarrollo de Aplicaciones Multiplataforma.

Alumno: Alejandro Garcia Pol.

Tutor: Luis Velasco Mena.

Proyecto: Tres en raya.

Índice

1.Justificación

Este proyecto comienza con la intención de aprender a desarrollar aplicaciones en Android, y termina convirtiéndose en el proyecto de final de curso.
Es por esto por lo que se parte desde un punto en el cual el proyecto esta ya en un estado avanzado, pero al haberlo comenzado a desarrollar teniendo escasos conocimientos de la materia, quedan muchas cosas que modificar, y muchos detalles que pulir sobre todo respecto a la forma de estructurar el codigo.

Este juego pretende agilizar la agudeza visual y la capacidad para resolver situaciones sencillas.

A diferencia de un tres en raya común, en este tablero solo se podrán utilizar tres fichas por jugador, teniendo que mover las fichas una vez se ha alcanzado este limite.
De esta forma se pretende aumentar la dificultad del juego, y evitar los empates, que son en el tres en raya común la mayoria de finales de partida.

2.Objetivos

Se tiene como objetivos principales de este proyecto:

-Mejorar el codigo y la inteligencia artificial del contrincante simulado.

-Añadir sonido.

-Crear una tabla de puntuaciones que se guarde en una base de datos del telefono.

-La aplicacion estara en inglés y español dependiendo de la configuración del telefono.

Lo más probable es que de tiempo a completar todos los objetivos, no obstante tratándose de programación, siempre pueden surgir nuevos problemas a resolver, que impidan la realización de alguno de ellos.

3.Desarrollo

El desarrollo comienza en un estado avanzado, en el cual tenemos un juego del tres en raya programado en android, utilizando el lenguaje java y el IDE eclipse.

Las características iniciales de las que se parten son:

-Un menú de navegación, mediante el cual podemos realizar diversas acciones:

Nueva partida: Juego del tres en raya por niveles jugando contra la misma aplicación.
Al terminar la partida, tanto si se pierde, como si se completa el juego se da al usuario la posibilidad de compartir su puntuación,en diversas redes sociales.
Para pasar cada nivel el jugador debera vencer a la cpu 3 veces, y solo podra perder el número de veces que se le indique antes de cada partida.
Y para pasar a la partida siguiente pulsara el botón "Jugar de Nuevo".

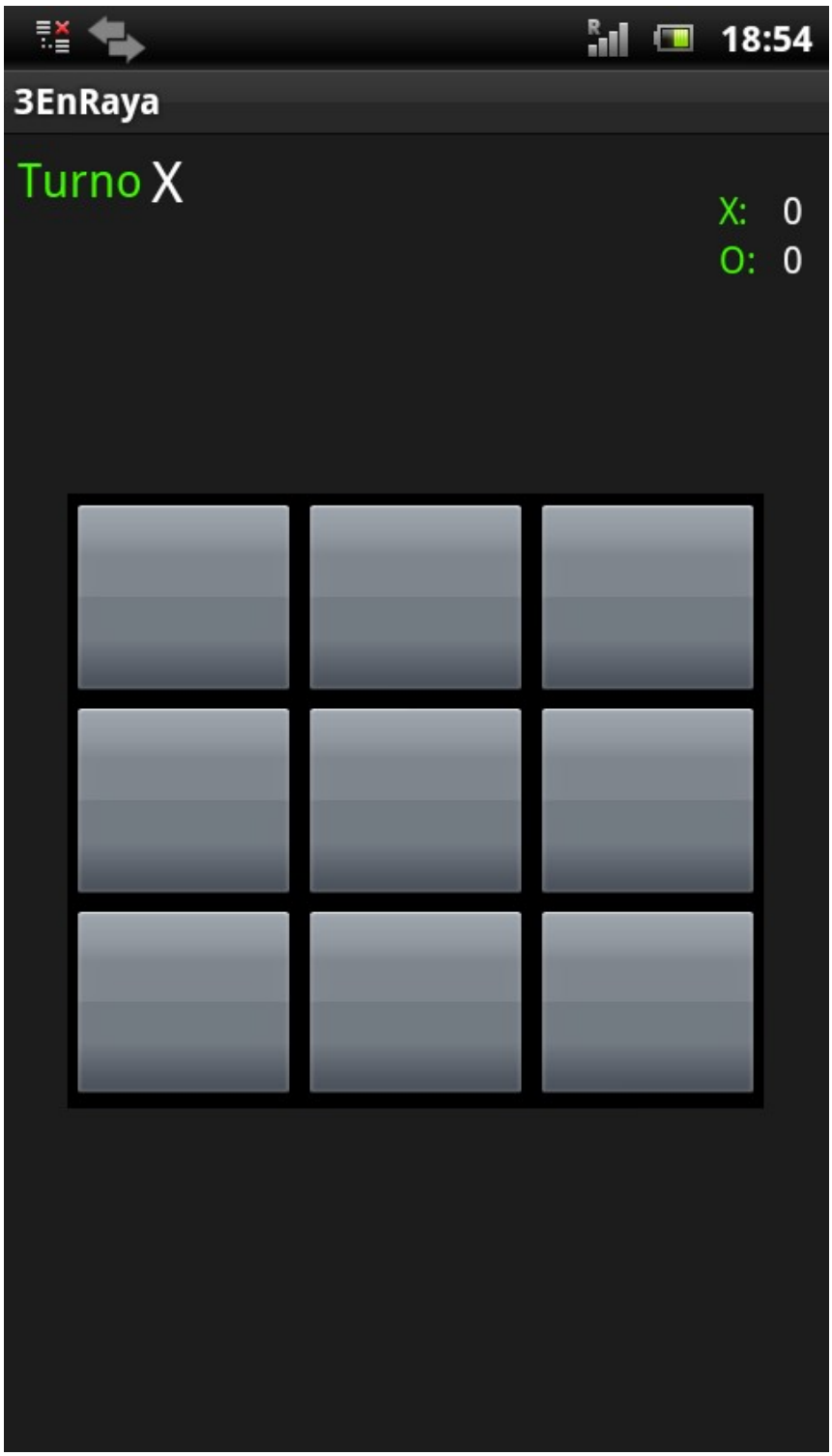

19:18
GameOver
GAME
OVER
Compartelo!
Puntos
-400


19:18
Completado
FELICIDADES
HAS
COMPLETADO
EL
JUEGO
Puntos 20700
Compartelo!

Salir: Salir de la aplicación para volver al escritorio de Android. Realmente no se cierra el proceso, sino que pasa a un segundo plano junto a otras aplicaciones que se van cerrando según el dispositivo necesita gestionar mayor cantidad de memoria.

Acerca de: Muestra un cuadro modal o dialogo, con información sobre la aplicación.

Multijugador: Partida por turnos dentro del mismo móvil entre dos personas. Sin puntuaciones.



Ya definido y explicado el desarrollo anterior al comienzo del proyecto vamos a profundizar en el desarrollo de los objetivos y la estructura del proyecto.

Limpieza

El primer paso que es necesario para comenzar el proyecto es limpiar las clases y paquetes innecesarios que habian quedado como restos de la versión anterior.
Puesto que en la anterior versión se habia hecho un intento de modo online quedan varios Layouts y Clases que corresponden a esta función así como cadenas del archivo Strings.xml contenedor de todas las cadenas de texto del juego tanto en Inglés como en Castellano.

Optimización del código

El código que controla los botones del tablero del tres en raya estan definidos uno a uno por separado, haciendo el código dificil de comprender, y muy costoso de modificar.
De modo que es necesario hacer unas modificaciones para solucionar este problema de código, convirtiéndo los botones en un array de botones pasandolo por un bucle for.

Código:

```
Button casillas[];

casillas = new Button[] {
          (Button)findViewById(R.id.Boton1),
          (Button)findViewById(R.id.Boton2),
          (Button)findViewById(R.id.Boton3),
          //fila2
          (Button)findViewById(R.id.Boton4),
          (Button)findViewById(R.id.Boton5),
          (Button)findViewById(R.id.Boton6),
          //fila3
          (Button)findViewById(R.id.Boton7),
          (Button)findViewById(R.id.Boton8),
          (Button)findViewById(R.id.Boton9),
```

```
        };




       for(index=0;index<=casillas.length-1;index++)//CONTROL DE BOTONES UNICO
       {
         casillas[index].setId(index);
        casillas[index].setOnClickListener(new OnClickListener()
        {
                  @Override
                  public void onClick(View v) {
                        Button b = (Button)v;
                  Debe_Contar = true;
                        if(Jugando)
                        {
                        CheckTurno();
                        if(b.getId()==0)
                        esta = "A";
                        if(b.getId()==1)
                        esta = "B";
                        if(b.getId()==2)
                        esta = "C";
                        if(b.getId()==3)
                        esta = "D";
                        if(b.getId()==4)
                        esta = "E";
                        if(b.getId()==5)
                        esta = "F";
                        if(b.getId()==6)
                        esta = "G";
                        if(b.getId()==7)
                        esta = "H";
                        if(b.getId()==8)
                        esta = "I";
                        if(pinta.equals(b.getText())||b.getText().equals(""))
                        {
                              if(pinta.equals("O") && OUsadas>=3 &&
b.getText().equals("")||pinta.equals("X") && XUsadas>=3 &&
b.getText().equals(""))//no devuelve nada cuando tienes las 3 fichas y pulsas un
boton vacio
                              {
                                    return;
                              }
PintaLlena(casillas[0],casillas[1],casillas[2],casillas[3],casillas[4],casillas[
5],casillas[6],casillas[7],casillas[8],again,ganador,b,XWins,OWins);


PintaVacia(casillas[0],casillas[1],casillas[2],casillas[3],casillas[4],casillas[
5],casillas[6],casillas[7],casillas[8],b);
                        if(b.getId()==0)
                        Ultima("A");
                        if(b.getId()==1)
                        Ultima("B");
                        if(b.getId()==2)
                        Ultima("C");
                        if(b.getId()==3)
                        Ultima("D");
```

```
                        if(b.getId()==4)
                        Ultima("E");
                        if(b.getId()==5)
                        Ultima("F");
                        if(b.getId()==6)
                        Ultima("G");
                        if(b.getId()==7)
                        Ultima("H");
                        if(b.getId()==8)
                        Ultima("I");
                        CuentaFichas();

EnableO(casillas[0],casillas[1],casillas[2],casillas[3],casillas[4],casillas[5],
casillas[6],casillas[7],casillas[8]);


EnableX(casillas[0],casillas[1],casillas[2],casillas[3],casillas[4],casillas[5],
casillas[6],casillas[7],casillas[8]);
                        ChangeTurno(jugador);
                        CheckTurno();


Cpu(casillas[0],casillas[1],casillas[2],casillas[3],casillas[4],casillas[5],casi
llas[6],casillas[7],casillas[8],again,ganador,b,XWins,OWins);
                        ChangeTurno(jugador);
                        }
                        }}});
       }
```

En la versión anterior teníamos el código siguiente para cada botón de la A a la I.

Código:

```
final Button A = (Button)findViewById(R.id.Boton1);
A.setOnClickListener(new OnClickListener()
        {
                  @Override
                  public void onClick(View v) {
                  Debe_Contar = true;
                        if(Jugando)
                        {
                        CheckTurno();
                        esta = "A";
                        if(pinta.equals(A.getText())||A.getText().equals(""))
                        {
                              if(pinta.equals("O") && OUsadas>=3 &&
A.getText().equals("")||pinta.equals("X") && XUsadas>=3 &&
A.getText().equals(""))//no devuelve nada cuando tienes las 3 fichas y pulsas un
boton vacio
                              {
                                    return;
                              }

      PintaLlena(A,B,C,D,E,F,G,H,I,again,ganador,A,XWins,OWins);
                        PintaVacia(A,B,C,D,E,F,G,H,I,A);
                        Ultima("A");
                        CuentaFichas();
```

```
                    EnableO(A,B,C,D,E,F,G,H,I);
                    EnableX(A,B,C,D,E,F,G,H,I);
                    ChangeTurno(jugador);
                    CheckTurno();
                    Cpu(A,B,C,D,E,F,G,H,I,again,ganador,A,XWins,OWins);
                    ChangeTurno(jugador);
                    }
                    }}});
```

Aprovechando esta mejora de código, optimizamos gran cantidad de metodos en toda la aplicación, al tener ahora mayor accesibilidad sobre los botones.

Una vez aplicado este cambio, procedemos a aplicarlo también a la clase multijugador que posee un código muy similar, pero sin implementar el metodo CPU.

Mejora de la inteligencia artificial

Dentro de este proyecto nos referimos a inteligencia artificial cuando hablamos de la forma de jugar y de responder a las jugadas que tiene la maquina cuando jugamos contra ella.

Esta inteligencia artificial creada a base de condiciones y comprobaciones consta de un ciclo de comprobación que se divide en x partes que se van recorriendo en el siguiente orden:

-Levantar una ficha(PintaVacia):

Comprueba si la maquina ya ha colocado las 3 fichas, y en caso de que así sea quita una ficha para más adelante colocarla en una casilla vacia. Este bloque no termina el ciclo ya que aun faltaría decidir donde se coloca la ficha que se ha quitado.

-Poner la ficha ganadora:

Comprueba que ya hay dos fichas puestas, y calcula si hay posibilidad de que la ficha que va a colocar a continuación forme tres en raya.
En caso de ser así coloca la ficha y gana la partida. En caso contrario pasa al siguiente bloque de comprobación.

-Tapar jugada del adversario:

Una vez comprobado que no se puede ganar en este movimiento, pasa a ser prioridad, el evitar que gane el adversario.
Para ello se comprueba si este tiene posibilidad de hacer tres en raya en el proximo movimiento.
En caso de ser así, se coloca la ficha en la casilla que evite la victoria del contrincante, y se pasa el turno saliendo del bucle.
Si no hay posibilidad de que el contrincante gane en el siguiente movimiento se pasa al siguiente bloque de comprobación.

-Colocar ficha aleatoria:

Por último, si todas las comprobaciones anteriores son negativas, se coloca la ficha en cualquier casilla vacia de forma aleatoria finalizando el bucle y pasando el turno al jugador humano.

Este bucle se llama cada vez que el jugador humano coloca una ficha, como respuesta a su jugada, y se va recorriendo hasta el final bloque por bloque hasta que encuentra una condición que le permite colocar la ficha y pasar el turno.

Una vez explicado el funcionamiento de la inteligencia artificial, pasemos a ver cual es el error que debemos corregir, para que las jugadas sean más efectivas, y resulte más complicado ganar a la máquina.

El fallo

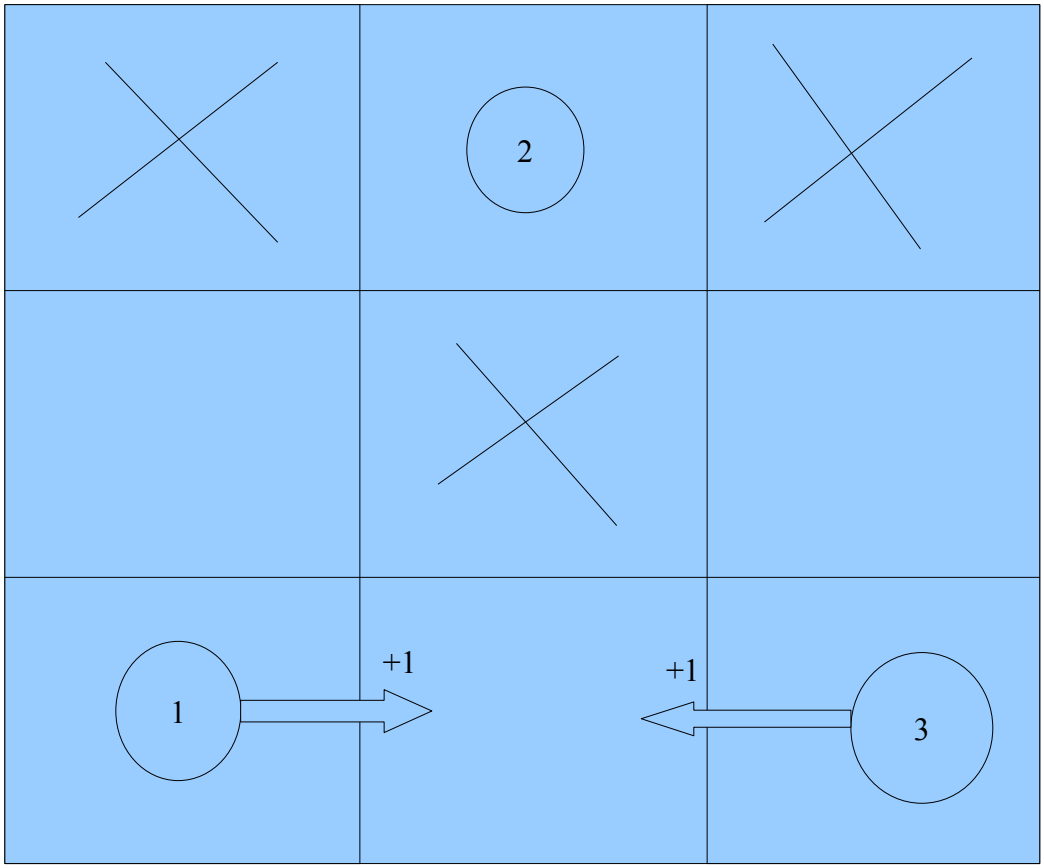
La cpu realiza respuestas lógicas, con intención de ganar en todas las situaciones.
El problema llega cuando tiene la posibilidad de hacer tres en raya, en la próxima jugada, pero ya tiene las tres fichas puestas, de modo que al levantar una y hacerlo de forma aleatoria, puede darse que levante una de las dos fichas implicadas en la jugada en lugar de levantar la que no lo esta y ponerla entre estas dos formando una línea.

De forma, que al calcular donde debe colocar la ficha, ya se ha roto la jugada, y se contempla otra situación diferente en la cual se procede a tapar la próxima jugada del adversario o poner una ficha aleatoria, cuando resultaba obvio que deberia haber ganado la partida.

La solución

La solución a este erróneo comportamiento, consiste en elaborar el bloque encargado de levantar una ficha, haciendo que no levante la ficha de forma aleatoria, sino siguiendo un criterio de comprobaciones que dictaminen cual es la ficha menos implicada en una jugada ganadora y levantarla.

Para ello una vez comprobado que tenemos las 3 fichas puestas en el tablero, comprobaremos las casillas de alrededor y sumaremos un punto por cada posibilidad de ganar en el siguiente movimiento, eligiendo finalmente la ficha con menos puntos que será levantada para colocarse posteriormente en la posición ganadora.



Ficha(1): 1 punto.
Ficha(2): 0 puntos.
Ficha(3): 1 punto.

En el ejemplo del dibujo, la ficha 1 tiene una posibilidad de ganar, igual que la ficha 3. Pero en cambio la ficha 2 no tiene ninguna, por lo tanto se levanta la ficha 2 colocándose entre la 1 y la 3 y corrigiendo el error antes mencionado.

Añadiendo sonidos

Con el propósito de mejorar la sensación de integración del jugador pasamos a introducir sonidos característicos que indiquen las acciones que ocurren en el juego a lo largo de la partida.

Los sonidos a programar son los siguientes:

-Sonido de colocación de ficha.

-Sonido de levantado de ficha.

-Sonido de casilla equivocada.

-Sonido de ganar la partida.

-Sonido de perder la partida.

Para implementar sonido en la aplicación, se ha utilizado la clase SoundPool, que es perfecta para sonidos cortos, y hace que ocupen poca memoria y puedan solaparse entre ellos, además de cambiar la frecuencia de muestreo(velocidad de reproducción del sonido), haciéndolo más personalizable.

A continuación, un ejemplo del codigo necesario para reproducir un sonido en android usando la clase SoundPool.

En la clase de la Activity:

```
private static SoundPool sound;
private static int Put;
```

En el metodo onCreate:

```
sound = new SoundPool(10,AudioManager.STREAM_MUSIC,0);
Put = sound.load(this, R.raw.put,1);
```

Donde queremos que suene(en este caso el metodo PintaVacia):

```
sound.play(Put, 1.0f, 1.0f, 0, 0, 1.5f);
```

El quinto parametro 1.5f, indica que el sonido se reproducira a una frecuencia de muestreo 0.5 veces mayor a la original(más rapido y en un tono más agudo).

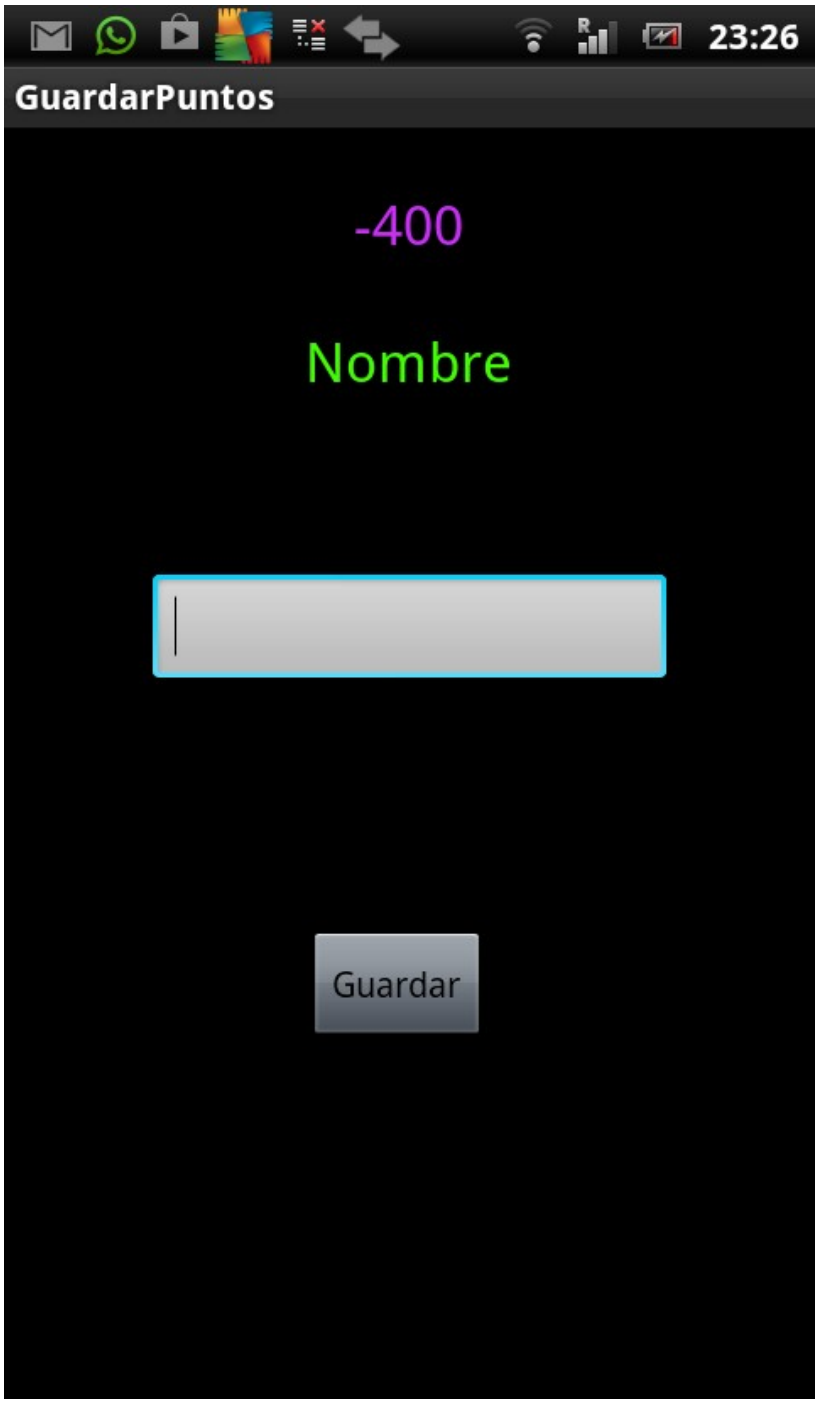
Si el valor fuese 1.0f sonaria en su frecuencia original.

En este caso he querido acortar el tiempo porque quedaba demasiado largo tratándose de el sonido correspondiente a una accion tan rápida como es colocar una ficha.

Añadiendo tabla de puntuaciones

Para crear la tabla de puntuaciones vamos a añadir un botón a la pantalla de game over, y a la pantalla de juego completado, junto al de compartir puntuación, el cual llevara a una nueva pantalla en la que se podrá escribir un nombre.

Esta pantalla tendrá un botón de guardar, el cual tras ser pulsado mostrara una lista con todas las puntuaciones que hayan sido guardadas en ese dispositivo, y las mostrara en orden descendente, mostrando primero el nombre del mejor jugador.




23:27
Score
Mejores Puntuaciones
loFuckinBoss 10050
flipone 3900
tgjretu 3350
alex 2833
alejandro 1550
alex 1250
cynthia 1250
vyuh 1250

Para implementar esta funcionalidad, crearemos
además varias clases.

Para crear los objetos en los que guardar el nombre y los puntos
creamos una clase llamada **Persona**, que acepta como parametros
un valor alfanumerico(el nombre), y un valor numerico(los puntos).

Necesitaremos también una clase helper, para mantener la base
de datos en la que almacenaremos la información(**UsuariosSQLiteHelper**).

Esta clase, se encarga de crear la tabla de puntuaciones, y de
actualizar sus valores cuando se detecten cambios.

Por último, la pantalla donde se visualizan las puntuaciones, en la cual
se muestra un ListView con los valores que se cargan de la tabla

Para cargar este ListView con las puntuaciones, se procede de la siguiente
manera:

-Se crea un ArrayList de objetos Persona, el cual se llena con los datos de la tabla score(puntuaciones), realizando una consulta a la base de datos.

-Se ordena de mayor a menor, respecto a las puntuaciones el ArrayList mediante un algoritmo de ordenación de fors anidados.

-Se le da estilo a los elementos de la lista.

-Se cargan los datos en la lista, mediante un ArrayAdapter.

Error al mostrar las puntuaciones

Una vez programadas las puntuaciones, procedemos, a comprobar su funcionalidad, y descubrimos un error.
Por algún motivo, los items de la lista aparecen duplicados.
Tras hacer varias comprobaciones, se confirma que la primera vez que se ejecuta la instalación recien instalada, no se da este fenómeno.
Pero que a medida que volvemos a jugar, y guardar puntuaciones los items de la lista se duplican de forma proporcional.

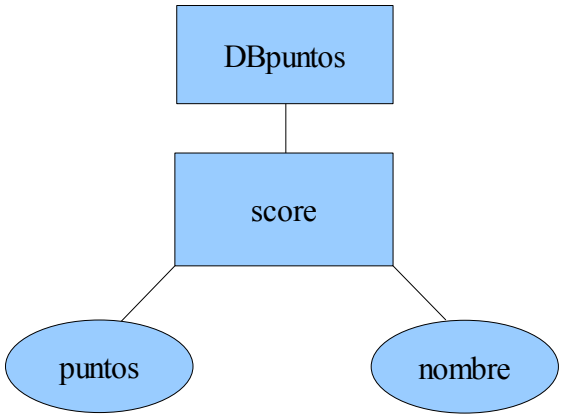
También se observa que contra más tiempo lleva la entrada creada en la tabla, mayor número de veces se duplica.

Al forzar el cierre de la aplicación vemos que no ocurre el fallo en la siguiente ejecución. Por lo que se llega a la conclusión final de que, la aplicación, al seguir ejecutándose en segundo plano esta guardando en la memoria volatil, el array con las puntuaciones, de manera que cada vez que se carga un nuevo valor, se duplica el contenido anterior del array.

Para solucionar este fallo, simplemente vaciamos el arraylist antes de volverlo a cargar con datos, evitando así duplicidades, y solucionando el imprevisto problema.

```
lista.clear();
```

Estructura base de datos puntuaciones



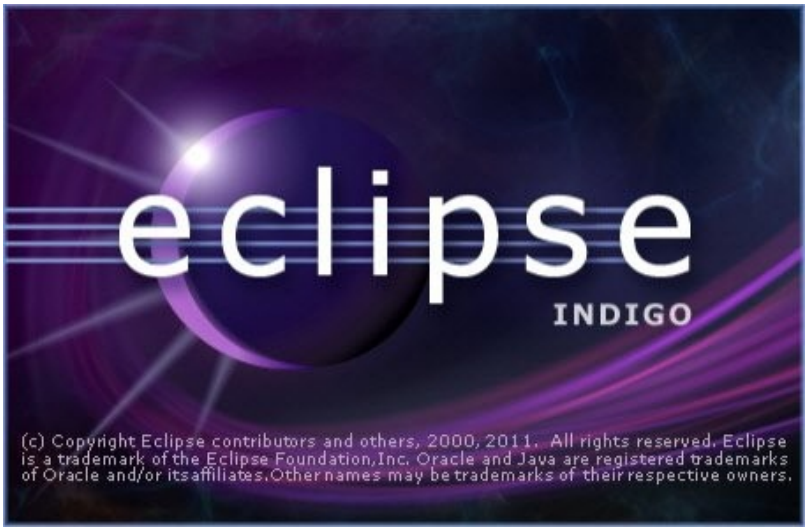
4.Tecnologia utilizada

Para la realización de este proyecto se han utilizado diversas herramientas:

El proyecto ha sido desarrollado para dispositivos Android, desde la versión 2.1 hasta la 4.2, usando el sdk de android que se encuentra alojado en los repositorios de google.

Como dispositivo de pruebas real, se ha utilizado un Sony Xperia Arc S.

La herramienta de desarrollo que se ha utilizado es el IDE Eclipse Indigo, junto al sdk de Android, en el cual se ha escrito todo el código de la aplicación, además de compilar las apk y realizar simulaciones en el emulador.



Para mantener el proyecto organizado, y llevar un control, tanto por parte del desarrollador, como por parte del tutor, se ha utilizado el control de versiones de git en la red publica github

El proyecto ha sido desarrollado tanto en pcs con Windows xp, como en ordenadores con Ubuntu, ya que el IDE Eclipse esta disponible para ambos sistemas operativos, y los proyectos se exportan facilmente.

5.Evaluación y conclusiones

Este proyecto podria dar mucho más de si, implementando múltiples caracteristicas, como el modo online, una mejora más refinada de la inteligencia artificial, y la mejora de la interfáz grafica, incluyendo recursos gráficos que le darian una apariencia más profesional de cara a un posible mercado.

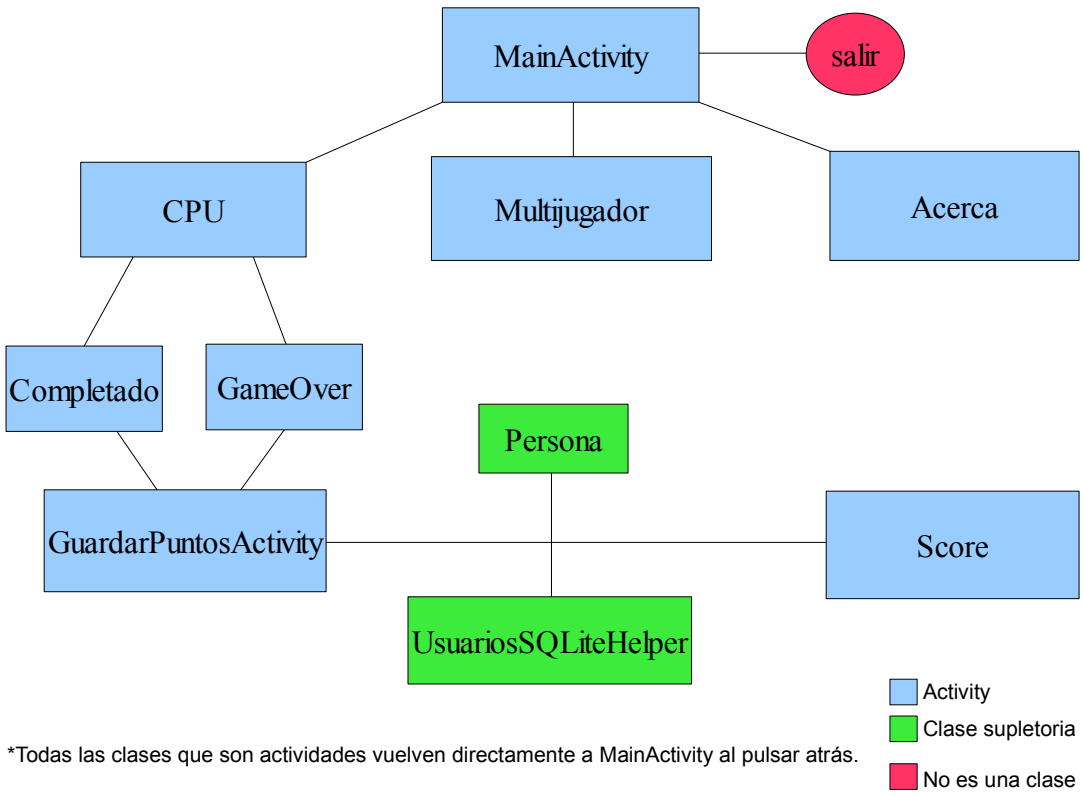
Pero no hay que olvidar que el proyecto esta limitado a unas horas/presupuesto por lo que habría que replantear más horas de trabajo, o pensar en programar una segunda versión de la aplicación de cara al futuro.

Finalmente, terminada la fase de desarrollo, se ha conseguido finalizar todos los objetivos que se habian planteado en la propuesta, por lo que se considera que el proyecto ha concluido con éxito dentro de sus limites establecidos, sin mayores problemas para realizar ninguna tarea.

El resultado del trabajo en este proyecto ha dado como fruto, un juego de tres en raya avanzado, funcional en todas las resoluciones de moviles y tabletas, disponible en dos idiomas(inglés y castellano), con modo multijugador y Contra la CPU, con la posibilidad de guardar las puntuaciones en una base de datos local, y de compartir las puntuaciones en las redes sociales,además de reproducir sonidos con las distintas acciones llevadas a cabo en el tablero de juego.

6.Anexo

Estructura de clases del proyecto:



*Todas las clases que son actividades vuelven directamente a MainActivity al pulsar atrás.

Icono de la aplicación:

Layout de la activity CPU(cpu.xml):

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#1C1C1C"
    android:gravity="clip_horizontal"
    tools:context=".CPU" >

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="46dp"
        android:layout_marginTop="26dp" >
    </RelativeLayout>

    <TableLayout
        android:id="@+id/tableLayout1"
        android:layout_width="270dp"
        android:layout_height="242dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="135dp"
        android:background="#000000" >

        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="3dp" >

            <Button
                android:id="@+id/Boton1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:minHeight="80dp"
                android:minWidth="90dp"
                android:text="@string/vacio"
                android:textSize="45dp" />
```

```
        <Button
            android:id="@+id/Boton2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />

        <Button
            android:id="@+id/Boton3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />
    </TableRow>

    <TableRow
        android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/Boton4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />

        <Button
            android:id="@+id/Boton5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />

        <Button
            android:id="@+id/Boton6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />
    </TableRow>

    <TableRow
        android:id="@+id/tableRow3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/Boton7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />

        <Button
            android:id="@+id/Boton8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />

        <Button
            android:id="@+id/Boton9"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minHeight="80dp"
            android:minWidth="90dp"
            android:text="@string/vacio"
            android:textSize="45dp" />
    </TableRow>
</TableLayout>

<Button
    android:id="@+id/again"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="60dp"
    android:text="@string/again" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/tableLayout1"
    android:layout_alignRight="@+id/again"
    android:layout_marginBottom="10dp"
    android:layout_marginRight="17dp"
    android:text="@string/vacio"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textColor="#40FF00" />

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="5dp"
    android:layout_marginTop="5dp"
    android:text="@string/turno"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="#40FF00" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginLeft="3dp"
        android:layout_marginTop="3dp"
        android:layout_toRightOf="@+id/textView1"
        android:text="@string/turno_def"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#FFFFFF" />

    <TableLayout
        android:id="@+id/tableLayout2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="10dp"
        android:layout_marginTop="20dp" >

        <TableRow
            android:id="@+id/tableRow5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/textView4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/X"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:textColor="#40FF00" />

            <TextView
                android:id="@+id/XWins"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:text="@string/contador_def"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:textColor="#FFFFFF" />
        </TableRow>

        <TableRow
            android:id="@+id/tableRow6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/TextView01"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/O"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:textColor="#40FF00" />

            <TextView
                android:id="@+id/OWins"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:text="@string/contador_def"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:textColor="#FFFFFF" />
```

```
        </TableRow>

        <TableRow
            android:id="@+id/tableRow7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
        </TableRow>

        <TableRow
            android:id="@+id/tableRow8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
        </TableRow>
    </TableLayout>

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textView2"
        android:layout_marginLeft="5dp"
        android:text="@string/level"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#40FF00"
        android:textSize="20sp" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView5"
        android:layout_alignBottom="@+id/textView5"
        android:layout_marginLeft="50dp"
        android:paddingLeft="7dp"
        android:text="1"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#ffffff" />

    <TextView
        android:id="@+id/score"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView7"
        android:layout_alignBottom="@+id/textView7"
        android:layout_toLeftOf="@+id/tableLayout2"
        android:text="0000"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#ffffff" />

    <TextView
        android:id="@+id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/again"
        android:layout_alignLeft="@+id/relativeLayout1"
        android:text="@string/score"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#40FF00" />

</RelativeLayout>
```

7.Referencias y bibliografia

Los conocimientos adquiridos para la realización de este proyecto han sido adquiridos en clase, de forma autodidacta en su mayoria, con la ayuda del IDE Eclipse, y de fuentes de internet.

La mayoría de consultas sobre código en la red han sido realizadas a http://stackoverflow.com

El codigo para pintar las letras de las puntuaciones de color verde ha sido extraido de stackoverflow:

```
@Override
    public View getView(int position, View convertView,//stackoverflow
            ViewGroup parent) {
        View view =super.getView(position, convertView, parent);

        TextView textView=(TextView) view.findViewById(android.R.id.text1);

        textView.setTextColor(Color.parseColor("#40FF00"));

        return view;
    }
};
```

Todos los sonidos utilizados en esta aplicación han sido obtenidos de forma gratuita en http://soundbible.com

Y en youtube http://www.youtube.com

Utilizando el programa atubeCatcher http://atube-catcher.dsnetwb.com/video/ que permite obtener media de youtube y convertir la salida a cualquier formato de audio/video.

Puedes seguir la evolución de este proyecto, colaborar en su desarrollo o simplemente descargarlo en https://github.com/AlejandroGarciaPol/3EnRaya

8.Instrucciones de uso

Nota: En este tres en raya cada jugador solo tiene 3 fichas para moverse por el tablero. Por lo tanto una vez se hayan colocado las tres fichas, se debe pulsar sobre una que ya este colocada, para quitarla de su sitio, y posteriormente pulsar sobre una casilla vacia para volver a colocarla.

1.Jugar contra la maquina:

Para jugar contra la máquina pulsaremos sobre el botón "Nueva Partida".
Inmediatamente aparecera un diálogo con las reglas de juego.
Para continuar al juego pulsaremos "ok".

En esta modalidad de juego siempre tiraremos nosotros primero y luego moverá ficha la maquina.
Nosotros seremos siempre las X, y la máquina las O.

Cada vez que se termine un tablero, pulsaremos sobre "Jugar de nuevo" para jugar la siguiente partida, y así hasta que perdamos todos los intentos o completemos los 10 niveles de juego.

Una vez completado el juego o perdido la partida, veremos una pantalla con nuestra puntuación, y dos botones. Si queremos compartir nuestra puntuación en las redes sociales pulsaremos sobre "compartir", y seleccionaremos la red social en la lista desplegable que aparecera.
Si lo que queremos es grabar nuestra puntuación en la tabla de records, pulsaremos sobre guardar, y se abrira una ventana para que introduzcamos nuestro nombre, posteriormente pulsaremos sobre guardar, y veremos la lista de records en la que podremos encontrar nuestra puntuación.

2.Jugar con un amigo:

Para jugar con otra persona pulsaremos sobre el botón "Multijugador".
Inmediatamente veremos el tablero con un contador de partidas ganadas para cada jugador.
El primer jugador que mueva sera las X, y el segundo las O.
Cada vez que un jugador gane la partida habra que pulsar en "jugar de nuevo", para resetear el tablero y volver a jugar.

3.Información sobre la aplicación:

En la pantalla principal, pulsar sobre "acerca de" para ver un cuadro de texto con información referente a la aplicación.
Pulsa la tecla de retroceso para cerrarla.

4.Salir de la aplicación, y navegar por las pantallas:

En la pantalla principal, pulsa sobre "salir", o sobre la tecla de retroceso.

Estando en cualquier pantalla, pulsa sobre la tecla de retroceso para ir a la pantalla principal.