

Funcionamiento:

La aplicación funciona con 4 Activities y 2 clases:

1 - MainActivity: Pantalla inicial que recoge un OBJETO PREFERENCIAS y que enlaza con el Layout principal.

Desde esta actividad se muestran las preferencias escogidas por el usuario y se accede a la pantalla de preferencias

2 - MainPreferencias: Actividad que genera y guarda un archivo SharedPreferences para las preferencias de la partida

3 – Clase PREFERENCIAS: Constructor de las diferentes preferencias de la partida usando un “aleatorizador” para construir tanto el número de bolas como la cantidad de colores de cada bola

4 – JuegoBolas: Activity que pinta todo el juego de bolas. Para el movimiento de las bolas he partido de un ejemplo resuelto de los apuntes, y para pintar el texto de preparación he usado una MAQUINA DE ESTADOS.

Esta máquina de estados pinta el texto de preparación, el temporizador y las bolas

Además hay un segundo temporizador invisible para pasar a la siguiente actividad.

5 - BolaParam: Sencillo constructor que construirá las bolas una vez tengamos las preferencias y numero de colores de cada bola con la clase PREFERENCIAS

6 -PantallaFin: Activity que comprueba los aciertos y fallos y reproduce un sonido especifico en caso de éxito o fracaso.

EMBOLATOR

EMBOLATOR



Preferencias de partida:

Número máximo de
colores: 3

Velocidad: Extrema

Max Bolas por cada color: 3

Tiempo: 10 segundos

Rebote: si

**CUENTA CUANTOS COLORES HAY DE
CADA BOLA!!**

COMENZAR

PREFERENCIAS

EMBOLATOR

VELOCIDAD

Extrema



**TIEMPO DE
PARTIDA**

10



Nº MAX DE BOLAS

3



Nº DE COLORES

3



ACTIVAR REBOTE



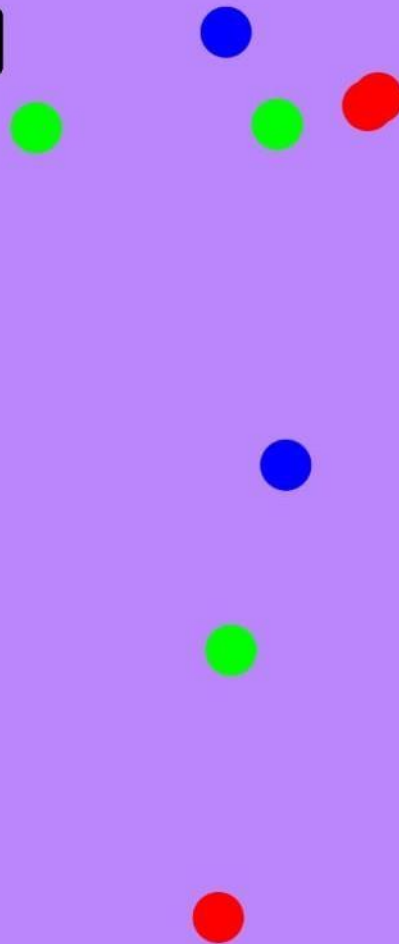
GUARDAR

EMBOLATOR

PREPARADOS

EMBOLATOR

10



EMBOLATOR

CUANTAS BOLAS HABIA DE CADA COLOR?

ROJO

3

VERDE

3

AZUL

2

AMARILLO

NEGRO

COMPROBAR

Colores acertados...

Capturas de código:

- Recuperación de referencias de la partida (archivo preferences) y envió a la actividad del juego

```
public void openJuego() {
    Intent intent = new Intent(this, JuegoBolas.class);

    preferences = getSharedPreferences("preferencias", Context.MODE_PRIVATE);
    p1 = preferences.getInt("velocidad", 20);
    p2 = preferences.getInt("tiempo", 20);
    p3 = preferences.getInt("maxBolas", 4);
    p4 = preferences.getInt("numColores", 4);
    p5 = preferences.getBoolean("rebote", true);

    p = new Preferencias(p1, p2, p3, p4, p5);

    intent.putExtra("p", p);
    startActivity(intent);
}
```

- Creación de un documento preferences para las preferencias de la partida

```
//inicializamos un objeto Editor para editar el documento de Preferencias
SharedPreferences.Editor editor = preferences.edit();
editor.putInt("velocidad", p1);
editor.putInt("tiempo", p2);
editor.putInt("maxBolas", p3);
editor.putInt("numColores", p4);
editor.putBoolean("rebote", p5);
editor.commit();

//Se inicializa una nueva actividad Main para dejar las preferencias fijadas en la pantalla de in
Intent intent = new Intent(this, MainActivity.class);
startActivity(intent);
```

- Randomizadores en clase Preferencias para el numero de colores

```
//Si numColores = 3, se aleatorizan rojas, verdes y azules, prestando atencion en el numero maximo de cada color que le hemos enviado desde la pantalla principal
if (numColores == 3) {
    numRojos = randomizador.nextInt(numMaxCadaColor) + 1;
    numVerdes = randomizador.nextInt(numMaxCadaColor) + 1;
    numAzules = randomizador.nextInt(numMaxCadaColor) + 1;
    numAmarillos = 0;
    numMagentas = 0;
    numBolas = numRojos + numVerdes + numAzules + numAmarillos + numMagentas;
}

//Si subimos en numero, vamos añadiendo colores
if (numColores == 4) {
    numRojos = randomizador.nextInt(numMaxCadaColor) + 1;
    numVerdes = randomizador.nextInt(numMaxCadaColor) + 1;
    numAzules = randomizador.nextInt(numMaxCadaColor) + 1;
    numAmarillos = randomizador.nextInt(numMaxCadaColor) + 1;
    numMagentas = 0;
    numBolas = numRojos + numVerdes + numAzules + numAmarillos + numMagentas;
}

if (numColores == 5) {
    numRojos = randomizador.nextInt(numMaxCadaColor) + 1;
    numVerdes = randomizador.nextInt(numMaxCadaColor) + 1;
    numAzules = randomizador.nextInt(numMaxCadaColor) + 1;
    numAmarillos = randomizador.nextInt(numMaxCadaColor) + 1;
    numMagentas = randomizador.nextInt(numMaxCadaColor) + 1;
    numBolas = numRojos + numVerdes + numAzules + numAmarillos + numMagentas;
}
```

- Montamos las bolas llevando información a BolasParam

```
//Dos array de enteros tanto para la direccion inicial (aleatoria) como para los diferentes colores que habra en el array de bolas
int[] coefVelo = new int[]{-1, 1};
int[] bolaColores = new int[numBolas];

//asignamos un numero a cada color segun el array COLORES, y montamos bolaColores segun las Preferencias
for (int j = 0; j < p.getNumRojos(); j++) {
    bolaColores[j] = 0;
}
for (int j = 0; j < p.getNumVerdes(); j++) {
    bolaColores[p.getNumRojos() + j] = 1;
}
for (int j = 0; j < p.getNumAzules(); j++) {
    bolaColores[p.getNumRojos() + p.getNumVerdes() + j] = 2;
}
for (int j = 0; j < p.getNumAmarillos(); j++) {
    bolaColores[p.getNumRojos() + p.getNumVerdes() + p.getNumAzules() + j] = 3;
}

for (int j = 0; j < p.getNumMagentas(); j++) {
    bolaColores[p.getNumRojos() + p.getNumVerdes() + p.getNumAzules() + p.getNumAmarillos() + j] = 4;
}

//Ahora si, inicializamos el array de bolas aleatorizando la dirección inicial
for (int i = 0; i < numBolas; i++) {
    int posX = randomizador.nextInt(1000);
    int posY = randomizador.nextInt(1000);
    int posCoefVelo = randomizador.nextInt(2);

    bolas[i] = new BolaParam(colores[bolaColores[i]], velocidad * coefVelo[posCoefVelo], velocidad * coefVelo[posCoefVelo], posX, posY, p.isRebote());
}
```


- Maquina de estados para la partida

```
//MAQUINA DE ESTADOS. Me servira para cambiar entre los distintos estados del juego.
switch (estado) {
    case 0:

        // PREPARADOS
        textPos += 30;
        c.drawText("PREPARADOS", (w / 2) - 300, h - textPos, pincelText);
        if (textPos >= h) {
            textPos = 0;
            estado++;
        }
        break; //Cambio de estados, pintamos siguiente texto
    case 1:
        // LISTOS
        textPos += 30;
        c.drawText("LISTOS", (w / 2) - 200, h - textPos, pincelText);
        if (textPos >= h) {
            textPos = getHeight();
            estado++;
        }
        break; //Cambio de estados, pintamos bolas
}
```

```
case 2:
    tick++;
    for (int i = 0; i < bolas.length; i++) {

        // Elegir color
        pincelBola.setColor(bolas[i].color);

        bolas[i].posX += bolas[i].velocidadX;
        bolas[i].posY += bolas[i].velocidadY;

        if (bolas[i].posX >= limiteDerecha) {
            if (bolas[i].rebote) {
                bolas[i].posX = limiteDerecha;
                bolas[i].velocidadX *= -1;
            } else {
                bolas[i].posX = 50;
            }
        }
        if (bolas[i].posX <= RADIO) {
            if (bolas[i].rebote) {
                bolas[i].posX = RADIO;
                bolas[i].velocidadX *= -1;
            } else {
                bolas[i].posX = limiteDerecha;
            }
        }
        if (bolas[i].posY >= limiteInferior) {
            if (bolas[i].rebote) {
                bolas[i].posY = limiteInferior;
                bolas[i].velocidadY *= -1;
            } else {
                bolas[i].posY = 50;
            }
        }
    }
}
```

```

        if (bolas[i].posY <= RADIO) {
            if (bolas[i].rebote) {
                bolas[i].posY = RADIO;
                bolas[i].velocidadY *= -1;
            } else {
                bolas[i].posY = limiteInferior;
            }
        }

        //Dos bucles if para el contador
        if (tick == 70) {
            tick = 0;
            segundos++;
        }

        if (segundos == tiempoTotal) {
            estado++;
        }
        // Dibujar el círculo
        c.drawCircle(bolas[i].posX, bolas[i].posY, RADIO, pincelBola);

        // Dibujar contador
        c.drawText(Integer.toString(tiempoTotal - segundos), 100, 200, pincelText);
    }
    break;
case 3: //Último estado, fin del juego

    break;

```

- Pantalla de Fin:

```
if (Integer.parseInt(rojo.getText().toString()) == p.getNumRojos()) {
    numAciertos++;
}

if (Integer.parseInt(verde.getText().toString()) == p.getNumVerdes()) {
    numAciertos++;
}
;
if (Integer.parseInt(azul.getText().toString()) == p.getNumAzules()) {
    numAciertos++;
}

if (Integer.parseInt(amarillo.getText().toString()) == p.getNumAmarillos()) {
    numAciertos++;
}

if (Integer.parseInt(magenta.getText().toString()) == p.getNumMagentas()) {
    numAciertos++;
}

//En caso de fallo, imprimir fallo y suena bocina
if (numAciertos != 5) {
    resultado.setText("Aciertos: " + numAciertos + " / " + "5");
    MediaPlayer mep = MediaPlayer.create(this, R.raw.sonido_fallaste);
    mep.start();
} else {
    resultado.setText("Aciertos: " + numAciertos + " / " + "5");
    MediaPlayer mep = MediaPlayer.create(this, R.raw.sonido_conseguido);
    mep.start();
    Intent intent = new Intent(this, MainActivity.class);
    startActivity(intent);
}
```