

## PRACTICA 3: ASTEROIDES XII

### Parte I: Fragmentando los asteroides

**Paso 1.** Convierte la variable local `drawableAsteroide` declarada en el constructor de la clase `VistaJuego`, en una variable global, que será un `array` de tres elementos:

`private Drawable drawableAsteroide[] = new Drawable[3];`

```
public VistaJuego(Context context, AttributeSet attrs) {
    super(context, attrs);

    //soundPool = new SoundPool( 5, AudioManager.STREAM_MUSIC, 0);
    soundPool = new SoundPool.Builder().setMaxStreams(5) .build();
    idDisparo = soundPool.load(context, R.raw.disparo, priority: 0);
    idExplosion = soundPool.load(context, R.raw.explosion, priority: 0);
    //mpDisparo= MediaPlayer.create(context,R.raw.disparo);
    //mpExplosion= MediaPlayer.create(context,R.raw.explosion);
    Drawable drawableNave, drawableAsteroide, drawableMisil;
```



```
private Drawable drawableAsteroide[]=new Drawable[3];

public VistaJuego(Context context, AttributeSet attrs) {
    super(context, attrs);

    //soundPool = new SoundPool( 5, AudioManager.STREAM_MUSIC, 0);
    soundPool = new SoundPool.Builder().setMaxStreams(5) .build();
    idDisparo = soundPool.load(context, R.raw.disparo, priority: 0);
    idExplosion = soundPool.load(context, R.raw.explosion, priority: 0);
    //mpDisparo= MediaPlayer.create(context,R.raw.disparo);
    //mpExplosion= MediaPlayer.create(context,R.raw.explosion);
    Drawable drawableNave, drawableMisil;
```

**Paso 2.** En el constructor cuando se quiera trabajar con bitmaps inicializaremos esta variable de la siguiente forma:

```
drawableAsteroide[0] = context.getResources().getDrawable(R.drawable.asteroide1);
drawableAsteroide[1] = context.getResources().getDrawable(R.drawable.asteroide2);
drawableAsteroide[2] = context.getResources().getDrawable(R.drawable.asteroide3);
```

```
} else{
    drawableAsteroide = ContextCompat.getDrawable(context, R.drawable.asteroide1);
    drawableNave = ContextCompat.getDrawable(context, R.drawable.nave);
    drawableMisil = ContextCompat.getDrawable(context, R.drawable.misil1);
}
```



```

} else{
    drawableAsteroide[0] = ContextCompat.getDrawable(context, R.drawable.asteroide1);
    drawableAsteroide[1] = ContextCompat.getDrawable(context, R.drawable.asteroide2);
    drawableAsteroide[2] = ContextCompat.getDrawable(context, R.drawable.asteroide3);
    drawableNave = ContextCompat.getDrawable(context, R.drawable.nave);
    drawableMisil = ContextCompat.getDrawable(context, R.drawable.misil1);
}

```

**Paso 3.** Y en caso de querer trabajar con gráficos vectoriales:

```

ShapeDrawable dAsteroide = new ShapeDrawable(
    new PathShape(pathAsteroide, stdWidth: 1, stdHeight: 1));
dAsteroide.getPaint().setColor(Color.WHITE);
dAsteroide.getPaint().setStyle(Paint.Style.STROKE);
dAsteroide.setIntrinsicWidth(50);
dAsteroide.setIntrinsicHeight(50);
drawableAsteroide = dAsteroide;

```



```

for (int i=0; i<3; i++) {
    ShapeDrawable dAsteroide = new ShapeDrawable(
        new PathShape(pathAsteroide, stdWidth: 1, stdHeight: 1));
    dAsteroide.getPaint().setColor(Color.WHITE);
    dAsteroide.getPaint().setStyle(Paint.Style.STROKE);
    dAsteroide.setIntrinsicWidth(50 - i * 14);
    dAsteroide.setIntrinsicHeight(50 - i * 14);
    drawableAsteroide[i] = dAsteroide;
}

```

**Paso 4.** Añade al principio del método `destruyeAsteroide(int i)` el código:

```

for (int p=0; p<Misiles.size();p++){
    if (misilesActivos.elementAt(p)){
        Misiles.elementAt(p).incrementaPos(retardo);
        tiempoMisiles.set(p, tiempoMisiles.get(p)-(int)retardo);
        if (tiempoMisiles.elementAt(p)<0){
            misilesActivos.set(p, false);
        }else{
            for (int i = 0; i < Asteroides.size(); i++) {
                if (Misiles.elementAt(p).verificaColision(Asteroides.elementAt(i))) {
                    Asteroides.remove(i);
                    misilesActivos.set(p, false);
                    //mpExplosion.start();
                    soundPool.play(idExplosion, leftVolume: 1, rightVolume: 0, priority: 0, loop: 0, rate: 2);
                    //soundPool.play(idExplosion, 1, 0, 0, 0, 2);
                    break;
                }
            }
        }
    }
}

```



```
for (int p = 0; p < Misiles.size(); p++) {
    if (misilesActivos.elementAt(p)) {
        Misiles.elementAt(p).incrementaPos(retardo);
        tiempoMisiles.set(p, tiempoMisiles.get(p) - (int) retardo);
        if (tiempoMisiles.elementAt(p) < 0) {
            misilesActivos.set(p, false);
        } else {
            int size = Asteroides.size();
            for (int i = 0; i < size; i++) {
                if (Misiles.elementAt(p).verificaColision(Asteroides.elementAt(i))) {
                    //DESTRUYE ASTEROIDE - CODIGO A AÑADIR
                    int tam;
                    if (Asteroides.get(i).getDrawable() != drawableAsteroide[2]) {
                        if (Asteroides.get(i).getDrawable() == drawableAsteroide[1]) tam = 2;
                        else tam = 1;
                    }
                    for (int n = 0; n < numFragmentos; n++) {
                        Grafico asteroide = new Grafico( View: this, drawableAsteroide[tam]);
                        asteroide.setPosX(Asteroides.get(i).getPosX());
                        asteroide.setPosY(Asteroides.get(i).getPosY());
                        asteroide.setIncX(Math.random() * 7 - 2 - tam);
                        asteroide.setIncY(Math.random() * 7 - 2 - tam);
                        asteroide.setAngulo((int) (Math.random() * 360));
                        asteroide.setRotacion((int) (Math.random() * 8 - 4));
                        Asteroides.add(asteroide);
                    }
                    //////////////////////////////////////
                    Asteroides.remove(i);
                    misilesActivos.set(p, false);
                    //mpExplosion.start();
                    soundPool.play(idExplosion, leftVolume: 1, rightVolume: 0, priority: 0, loop: 0, rate: 2);
                    //soundPool.play(idExplosion, 1, 0, 0, 0, 2);
                    break;
                }
            }
        }
    }
}
```

Si hay colision y tiene el tamaño 2 (el más pequeño), directamente es eliminado. Si tiene tamaño 1 ó 0, creará numFragmentos asteroides de tamaño 2 ó 1 respectivamente, y posteriormente será eliminado.

**Paso 5.** Corrige algún error adicional ocasionado por este cambio.



```

//////////2. CONSTRUCTOR//////////
nave = new Grafico( view: this, drawableNave);
Asteroides = new Vector<Grafico>();
for (int i = 0; i < numAsteroides; i++) {
    Grafico asteroide = new Grafico( view: this, drawableAsteroide[0]);
    asteroide.setIncY(Math.random() * 4 - 2);
    asteroide.setIncX(Math.random() * 4 - 2);
    asteroide.setAngulo((int) (Math.random() * 360));
    asteroide.setRotacion((int) (Math.random() * 8 - 4));
    Asteroides.add(asteroide);
}

```

**Paso 6.** Prueba los cambios propuestos.



## Parte II: Mejorando preferencias en Asteroides

**Paso 1.** Modifica el programa para que el número de fragmentos generados corresponda con el valor introducido en las preferencias.

```

///// ASTEROIDES /////
private Vector<Grafico> Asteroides; // Vector con Los Asteroides
private int numAsteroides= 5; // Número inicial de asteroides
private int numFragmentos= 3; // Fragmentos en que se divide

```

```

<ListPreference
    android:key="graficos"
    android:title="Tipo de gráficos"
    android:summary="Se escoge la representación de gráficos"
    android:entries="@array/tiposGraficos"
    android:entryValues="@array/tiposGraficosValores"
    android:defaultValue="1"/>
<EditTextPreference
    android:key="fragmentos"
    android:title="Número de Fragmentos"
    android:summary="En cuantos trozos se divide un asteroide"
    android:inputType="number"
    android:defaultValue="3"/>

```

```
SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(getContext());  
numFragmentos = Integer.parseInt(pref.getString( key: "fragmentos", defValue: "3"));  
  
if (pref.getString( key: "graficos", defValue: "1").equals("0")) {
```

**Paso 2.** Puedes aprovechar para que la reproducción de música de fondo y los efectos de audio sean también configurables por el usuario.