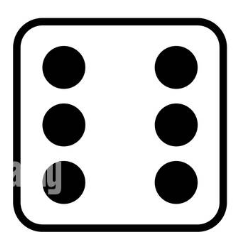
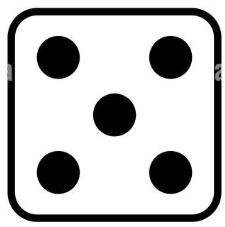
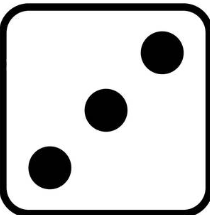
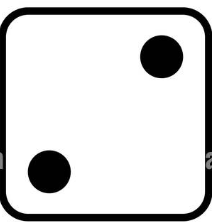
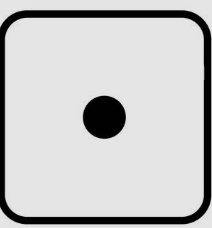
# Aplikacija- Shake the dice



**package** com.example.myapplication16;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Context;  
**import** android.hardware.Sensor;  
**import** android.hardware.SensorEvent;  
**import** android.hardware.SensorEventListener;  
**import** android.hardware.SensorManager;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.view.Window;  
**import** android.widget.ImageView;  
**import** android.widget.Toast;  
  
**import** java.util.Random;  
  
**public class** MainActivity **extends** AppCompatActivity **implements** SensorEventListener {  
  
 **private** SensorManager **senSensorManager**;  
 **private** Sensor **senAccelerometer**;  
  
 ImageView **imageView**;  
 Random **r**;  
 **float x**, **y**, **z**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 requestWindowFeature(Window.***FEATURE\_NO\_TITLE***);  
 getSupportActionBar().hide();  
  
 setContentView(R.layout.***activity\_main***);  
  
 **senSensorManager** = (SensorManager) getSystemService(Context.***SENSOR\_SERVICE***);  
 **senAccelerometer** = **senSensorManager**.getDefaultSensor(Sensor.***TYPE\_ACCELEROMETER***);  
 **senSensorManager**.registerListener(**this**, **senAccelerometer** , SensorManager.***SENSOR\_DELAY\_NORMAL***);  
  
  
 **imageView** = findViewById(R.id.***imageView***);  
 ImageView imageView = **new** ImageView(**this**);  
 }  
  
 @Override  
 **public void** onSensorChanged(SensorEvent event) {  
 **if** (event.**sensor**.getType() == Sensor.***TYPE\_ACCELEROMETER***) {  
 **float**[] values = event.**values**;  
 *// Movement* **float** x = values[0];  
 **float** y = values[1];  
 **float** z = values[2];  
  
 **float** aSquareRoot = (x \* x + y \* y + z \* z)  
 / (SensorManager.***GRAVITY\_EARTH*** \* SensorManager.***GRAVITY\_EARTH***);  
 **if** (aSquareRoot >= 2) *//it will be executed if you shuffle* {  
  
 Random r = **new** Random();  
 **int**[] imageList = **new int**[]{R.drawable.***kocka1***, R.drawable.***kocka2***, R.drawable.***kocka3***, R.drawable.***kocka4***, R.drawable.***kocka5***, R.drawable.***kocka6***};  
 **imageView**.setImageResource(imageList[r.nextInt(imageList.**length**)]);  
  
 }  
  
 }  
 }  
  
 @Override  
 **public void** onAccuracyChanged(Sensor sensor, **int** accuracy) {  
  
 }  
  
  
}

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.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"**>  
  
 <**ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="68dp"  
 android:layout\_marginTop="72dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/kocka1"** />  
</**androidx.constraintlayout.widget.ConstraintLayout**>

# Aplikacija – Bounce Ball hits the frames

**package** com.example.myapplication16;  
  
**import** android.content.Context;  
**import** android.hardware.Sensor;  
**import** android.hardware.SensorEvent;  
**import** android.hardware.SensorEventListener;  
**import** android.hardware.SensorManager;  
**import** android.media.AudioManager;  
**import** android.media.ToneGenerator;  
**import** android.os.Vibrator;  
  
**import** android.os.Bundle;  
**import** android.util.DisplayMetrics;  
**import** android.widget.ImageView;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**public class** MainActivity **extends** AppCompatActivity **implements** SensorEventListener{  
  
 *//Sensor* **private** SensorManager **sensorManager**;  
 **private** Sensor **gravitySensor**;  
  
 *//The ball to move around on the screen* **private** ImageView **ball**;  
  
 *//The width of the frame* **private int frameWidth**;  
 *//The height of the frame* **private int frameHeight**;  
 *//Used to make the ball stay within the frame* **private int radius**;  
  
 *//Used to play a sound when the ball hits the edge of the frame* **private** ToneGenerator **toneGenerator**;  
  
 *//Gets the screens height and width* DisplayMetrics **displayMetrics**;  
 **private int devHeight**;  
 **private int devWidth**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *//Gets the frame and the ball* ImageView frame = findViewById(R.id.***frame***);  
 **ball** = findViewById(R.id.***ball***);  
  
 *//Initiates a new ToneGenerator* **toneGenerator** = **new** ToneGenerator(AudioManager.***STREAM\_MUSIC***, ToneGenerator.***MAX\_VOLUME***);  
  
 *//Sets the height and width of the frame* **frameWidth** = frame.getWidth();  
 **frameHeight** = frame.getHeight();  
  
 *//Sets the radius* **radius** = 25;  
  
 *//Gets the system sensors* **sensorManager** = (SensorManager)getSystemService(***SENSOR\_SERVICE***);  
 **assert sensorManager** != **null**;  
 *//Selects the gravitySensor sensor* **gravitySensor** = **sensorManager**.getDefaultSensor(Sensor.***TYPE\_GRAVITY***);  
  
 *//Gets the screens height and width* **displayMetrics** = **new** DisplayMetrics();  
 getWindowManager().getDefaultDisplay().getMetrics(**displayMetrics**);  
 **devHeight** = **displayMetrics**.**heightPixels**;  
 **devWidth** = **displayMetrics**.**widthPixels**;  
  
 *//Creates a listener on the sensor* **sensorManager**.registerListener(**this**, **gravitySensor**, SensorManager.***SENSOR\_DELAY\_GAME***);  
 *//Set the delay of to GAME* }  
  
 @Override  
 **public void** onSensorChanged(SensorEvent sensorEvent) {  
 *//Current coordinates of the ball* **float** X = **ball**.getX();  
 **float** Y = **ball**.getY();  
  
 *//The coordinates the ball moves to* **float** nextX = X + sensorEvent.**values**[1];  
 **float** nextY = Y + sensorEvent.**values**[0];  
  
 *//moves the ball LEFT until it crashes with the border* **if**((nextX - **radius**) >= **frameWidth** / 2){  
 **ball**.setX(nextX);  
 }**else** {  
 onFrameHit();  
 *//"Bounces" the ball of the frame* **ball**.setX(nextX + 35);  
 }  
 *//moves the ball to the TOP until it crashes with the border  
 //Added "+ 20" to make the ball bounce on the inner edge of the frame* **if**((nextY - **radius**) >= (**frameHeight** / 2) + 20){  
 **ball**.setY(nextY);  
 }**else**{  
 onFrameHit();  
 **ball**.setY(nextY + 35);  
 }  
  
 *//moves the ball to the RIGHT until it crashes with the border  
 //Added "- 60" to make the ball bounce on the frame and not the edge of the screen* **if** ((nextX + **radius**) > **devWidth** - (**frameWidth** / 2) - 180) {  
 onFrameHit();  
 **ball**.setX(nextX - 35);  
 }  
  
 *//moves the ball to the BOTTOM until it crashes with the border  
 //Added "- 320" to make the ball bounce on the frame and off the edge of the screen* **if** ((nextY + **radius**) > **devHeight** - (**frameHeight** / 2) - 420) {  
 onFrameHit();  
 **ball**.setY(nextY - 35);  
 }  
  
 }  
  
 @Override  
 **public void** onAccuracyChanged(Sensor sensor, **int** i) {  
 *//NOT USED, but implemented as SensorEventListener needs it* }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 *//Creates a listener on the gravity-sensor on resuming the app* **sensorManager**.registerListener(**this**, **gravitySensor**, SensorManager.***SENSOR\_DELAY\_GAME***);  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 *//Removes the listener when the app is not in use* **sensorManager**.unregisterListener(**this**);  
 }  
  
 */\*\*  
 \* Sets of multiple effects when the ball hits the edge of the frame  
 \*/* **private void** onFrameHit(){  
 *//Vibrator* Vibrator vibrator = (Vibrator) getSystemService(Context.***VIBRATOR\_SERVICE***);  
 **assert** vibrator != **null**;  
 vibrator.vibrate(400);  
  
 *//Sound* **toneGenerator**.startTone(ToneGenerator.***TONE\_CDMA\_ONE\_MIN\_BEEP***);  
 }  
}

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.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"**>  
  
 <**ImageView  
 android:id="@+id/frame"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="27dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="27dp"  
  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@color/teal\_700"** />  
  
 <**ImageView  
 android:id="@+id/ball"  
 android:layout\_width="67dp"  
 android:layout\_height="76dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintStart\_toStartOf="@+id/frame"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.499"  
 app:srcCompat="@android:drawable/radiobutton\_off\_background"** />  
</**androidx.constraintlayout.widget.ConstraintLayout**>

<https://www.journaldev.com/8988/android-studio-tutorial-hello-world-app>

Zadatak: kada loptica dodiruje lijevu stranu okvira ima žutu boju kada je na desnoj ima crvenu boju, kada je na vrhu okvira ima bijelu boju, a kada je na donjem dijelu okvira mijenja boju u crnu.

Vježba:

Napraviti formu: unosi ime korisnika, prezime korisnika, e-mail, radiobutton – spol (M ili F). Otvara novu aktivnost u kojoj to piše sve na jednom mjestu.