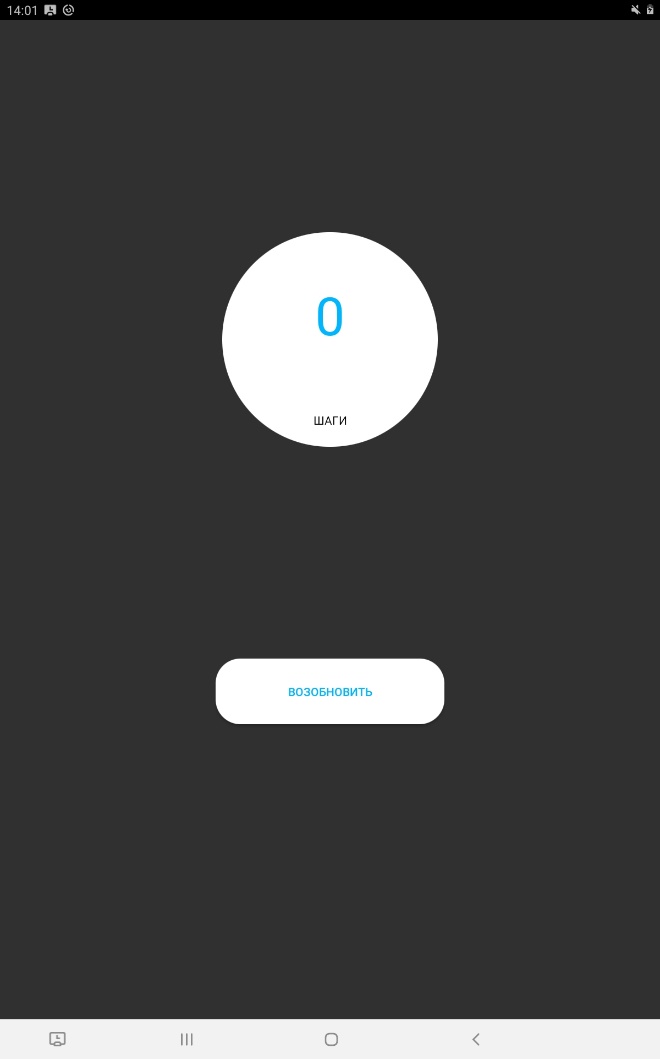
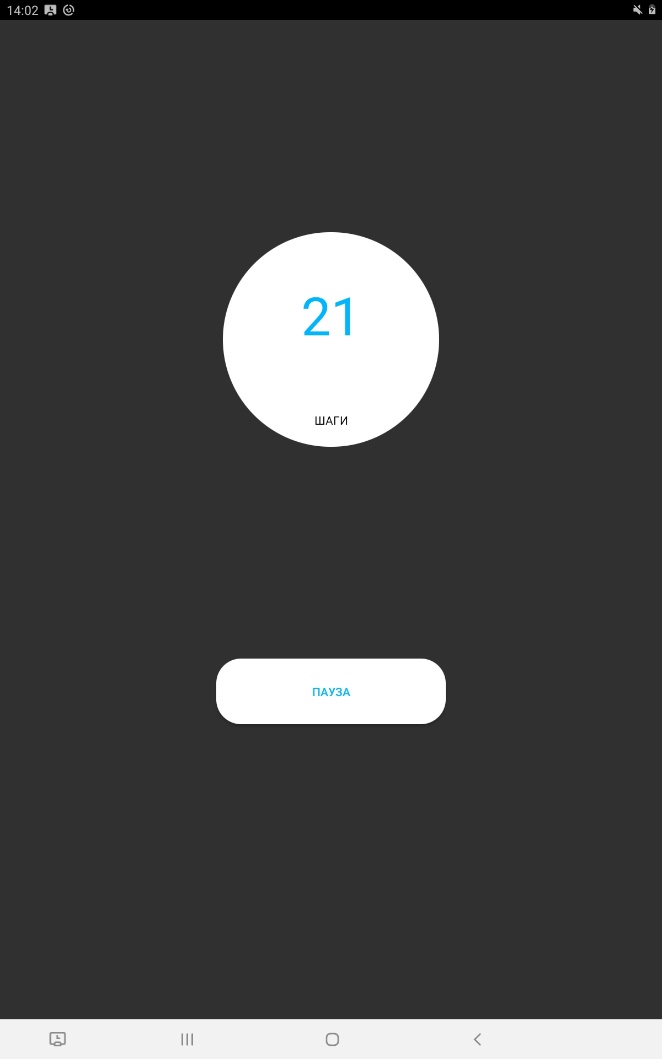
Внешний вид приложения:

Код:

Activity\_main.xml:

<?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">  
  
 <!--  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 -->  
  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="261dp"  
 android:layout\_height="259dp"  
 android:layout\_marginVertical="0dp"  
 android:layout\_marginStart="70dp"  
 android:layout\_marginTop="192dp"  
 android:layout\_marginEnd="70dp"  
 android:layout\_marginBottom="32dp"  
 android:background="@drawable/radius"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toTopOf="@+id/button"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="195dp"  
 android:gravity="center"  
 android:text="0"  
 android:textColor="#00B7FF"  
 android:textSize="64sp" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="64dp"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text="ШАГИ"  
 android:textColor="#000000" />  
  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="277dp"  
 android:layout\_height="79dp"  
 android:layout\_marginBottom="100dp"  
 android:background="@drawable/radius\_button"  
 android:onClick="OnStoped"  
 android:text="ПАУЗА"  
 android:textColor="#00B7FF"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/linearLayout" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.java:

package com.addon.pedometer;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
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.widget.Button;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity implements SensorEventListener {  
  
 public boolean active = true;  
 private SensorManager sensorManager; //Объект для работы с датчиком  
 private int count = 0; //Количество шагов  
 private TextView text; //Ссылка на TextView  
 private long lastUpdate; //Время последнего изменения состояния датчика  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 text = findViewById(R.id.*textView2*); //Находим текст  
 text.setText(String.*valueOf*(count)); //Создаем объект, для работы с датчиком  
 sensorManager=(SensorManager) getSystemService(*SENSOR\_SERVICE*); //Регистрируем класс, где будет реализован метод, вызываемый при изменении  
  
 //Запускаем датчик  
 sensorManager.registerListener((SensorEventListener) this,  
 sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*),  
 SensorManager.*SENSOR\_DELAY\_NORMAL*);  
  
 lastUpdate = System.*currentTimeMillis*(); //Устанавливаем последнее время обновления  
  
  
 Button button = findViewById(R.id.*button*);  
 button.setText("Пауза");  
 }  
  
 @Override  
 protected void onResume(){ //Продолжить  
 super.onResume(); //Подписываемся на действие  
 sensorManager.registerListener((SensorEventListener) this,  
 sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*),  
 SensorManager.*SENSOR\_DELAY\_NORMAL*); //Запускаем датчик  
 }  
  
 @Override  
 protected void onPause(){ //Пауза  
 super.onPause(); //Подписываемся на действие  
 sensorManager.unregisterListener((SensorEventListener) this); //Останавливаем датчик  
 }  
  
 //Функция паузы  
 public void OnStoped(View view) {  
 active = !active; //Активно/неактивно  
 if(!active) { //Если неактивно  
 Button button = findViewById(R.id.*button*); //Находим кнопку  
 button.setText("Возобновить"); //Присваиваем текст  
 } else { //Если активно  
 Button button = findViewById(R.id.*button*); //Находим кнопку  
 button.setText("Пауза"); //Присваиваем текст  
 }  
 }  
  
 public void onSensorChanged(SensorEvent event){ //Функция изменения датчика  
 Button button = findViewById(R.id.*button*);  
 if(button.getText() == "Пауза")  
 {  
 if(event.sensor.getType() == Sensor.*TYPE\_ACCELEROMETER*) { //Если акселерометра  
 float[] values = event.values; //Проекции ускорения на оси системы координат  
 float x = values[0]; //Координата x  
 float y = values[1]; //Координата y  
 float z = values[2]; //Координата z  
  
 //Квадрат модуля ускорения телефона, деленный на квадрат ускорения свободного падения  
 float accelationSquareRoot = (x\* x+y \* y+z \*z) / (SensorManager.*GRAVITY\_EARTH*\*SensorManager.*GRAVITY\_EARTH*);  
  
 //Текущее время  
 long actualTime=System.*currentTimeMillis*();  
  
 if(accelationSquareRoot>=2){ //Если тряска сильная  
 if(actualTime - lastUpdate<200){ //Если с момента начала тряски прошло меньше 200 миллисекунд - выходим из обработчика  
 return;  
 }  
 lastUpdate = actualTime; //Актуализируем данные  
  
 count++; //Увеличиваем шаг  
 text.setText(String.*valueOf*(count)); //Обновляем текст  
 }  
 }  
 }  
  
 }  
  
 //Метод сообщения о движении датчика с разной скоростью  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int accuracy) {  
  
 }  
}

radius.xml:

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">  
  
 <solid android:color="#FFFFFF"/>  
  
 <stroke android:width="0dp"  
 android:color="#ff000000"/>  
  
 <padding android:left="2dp"  
 android:right="2dp"  
 android:top="2dp"  
 android:bottom="2dp"/>  
  
 <corners android:radius="135dp"/>  
</shape>

Radius\_button.xml:

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">  
  
 <solid android:color="#FFFFFF"/>  
  
 <stroke android:width="0dp"  
 android:color="#ff000000"/>  
  
 <padding android:left="2dp"  
 android:right="2dp"  
 android:top="2dp"  
 android:bottom="2dp"/>  
  
 <corners android:radius="30dp"/>  
</shape>