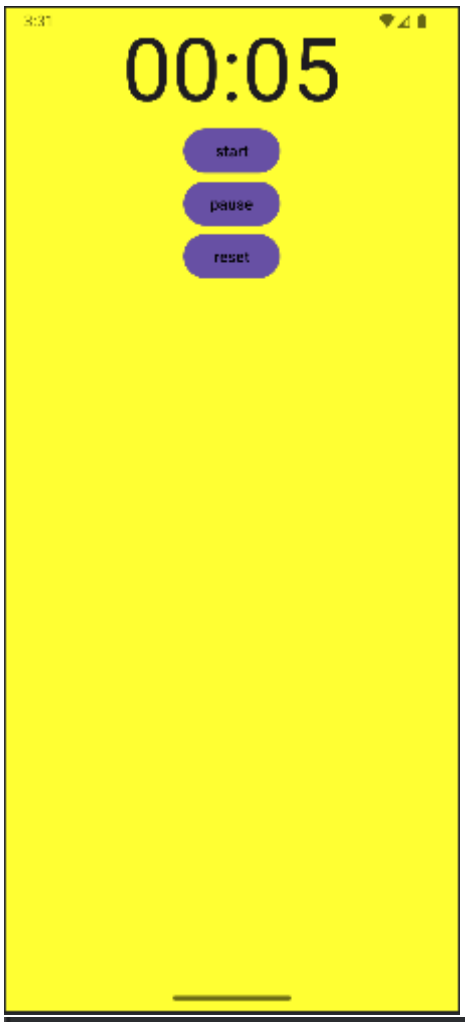

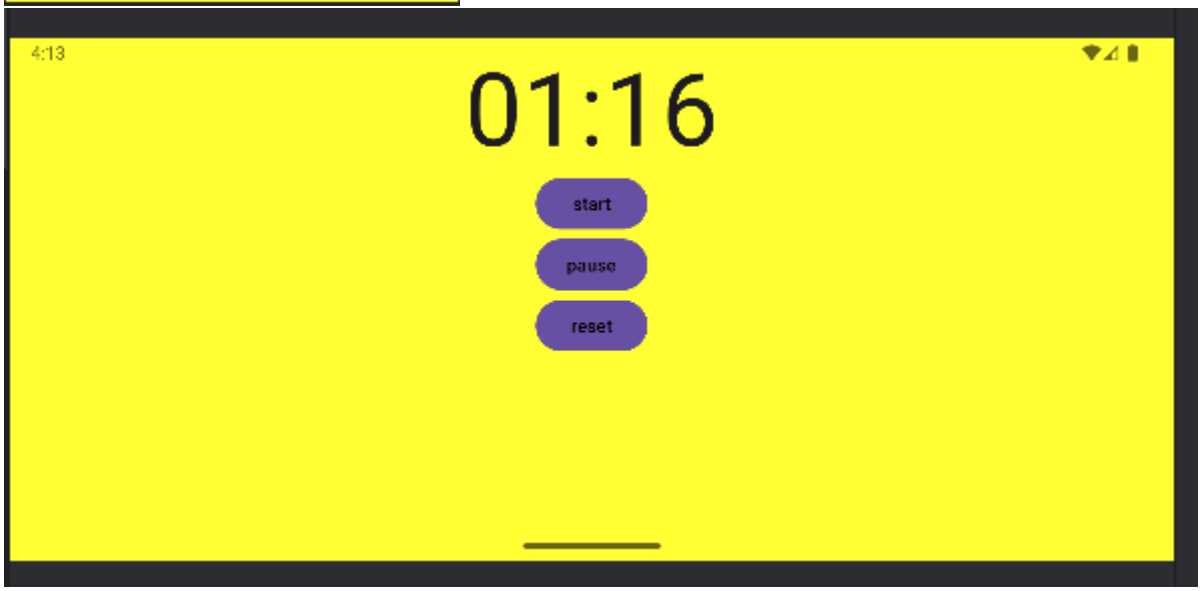

Activity_main.xml

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/yellow"
    tools:context=".MainActivity">

    <Chronometer
        android:id="@+id/textTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="80sp"
        android:layout_gravity="center"
        />
    <Button
        android:id="@+id/btnStart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/start"
        android:layout_gravity="center"
        android:textColor="@color/black"
        />
    <Button
        android:id="@+id/btnPause"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/pause"
        android:layout_gravity="center"
        android:textColor="@color/black"
        />
    <Button
        android:id="@+id/btnReset"
        android:layout_width="wrap_content"
```

```
        />
    <Button
        android:id="@+id/btnReset"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/reset"
        android:layout_gravity="center"
        android:textColor="@color/black"
        />
</LinearLayout>
```

String.xml

```
<resources>
    <string name="start">start</string>
    <string name="pause">pause</string>
        <string name="reset">reset</string>
    <string name="app_name">string\n\n</string>
</resources>
```

MainActivity.kt

```
package com.example.chronometr

import android.os.Bundle
import android.os.SystemClock
import android.widget.Button
import android.widget.Chronometer
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
    lateinit var chronometer: Chronometer
    var running = false
    var offset: Long = 0

    // Константы для ключей в Bundle
    private val OFFSET_KEY = "offset"
    private val RUNNING_KEY = "running"
    private val BASE_KEY = "base_key"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)

        // Инициализация элементов View
        chronometer = findViewById(R.id.textTime)
        val btnStart = findViewById<Button>(R.id.btnStart)
        val btnPause = findViewById<Button>(R.id.btnPause)
        val btnReset = findViewById<Button>(R.id.btnReset)

        // Обработка нажатий
        btnStart.setOnClickListener {
            if (!running) {
                setBaseTime()
                chronometer.start()
```

```
                running = true
            }
        }

        btnPause.setOnClickListener {
            if (running) {
                saveOffset()
                chronometer.stop()
                running = false
            }
        }

        btnReset.setOnClickListener {
            offset = 0
            setBaseTime()
            running = false
        }

        // Восстановление состояния при создании активности (например, после поворота)
        if (savedInstanceState != null) {
            offset = savedInstanceState.getLong(OFFSET_KEY)
            running = savedInstanceState.getBoolean(RUNNING_KEY)
            chronometer.base = savedInstanceState.getLong(BASE_KEY)
            if (running) {
                chronometer.start()
            } else {
                setBaseTime()
            }
        }
    }

    // Сохранение состояния перед уничтожением активности
    override fun onSaveInstanceState(outState: Bundle) {
```

```
    // Сохранение состояния перед уничтожением активности
    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        outState.putBoolean(RUNNING_KEY, running)
        outState.putLong(OFFSET_KEY, offset)
        outState.putLong(BASE_KEY, chronometer.base)
    }

    private fun saveOffset() {
        offset = SystemClock.elapsedRealtime() - chronometer.base
    }

    private fun setBaseTime() {
        chronometer.base = SystemClock.elapsedRealtime() - offset
    }
}
```