Content:

1.Introduction...............................................................................................................................................1 1.1 Objectives..........................................................................................................................................1 1.2 Specifications................................................................................................................................1 1.3 2.Analisys......................................................................................................................................................3 2.1 Multiplication......................................................................................................................................3 2.2 Division................................................................................................................................................4 4.Design ........................................................................................................................................................5 5.Implementation.........................................................................................................................................7 6.Testing and validation...............................................................................................................................8 7.Conclusions................................................................................................................................................9 8.Bibliography ............................................................................................................................................10

**1.Introduction**

1.1 Objectives

This paper is about making a Benchmarking app for mobile phones. We want to build an app that tests how good a phone is in three ways:

* How fast the CPU (the brain of the phone) works;
* How well the GPU (what makes graphics look good) performs;
* How the memory (where the phone stores data) handles tasks.

1.2 Specifications

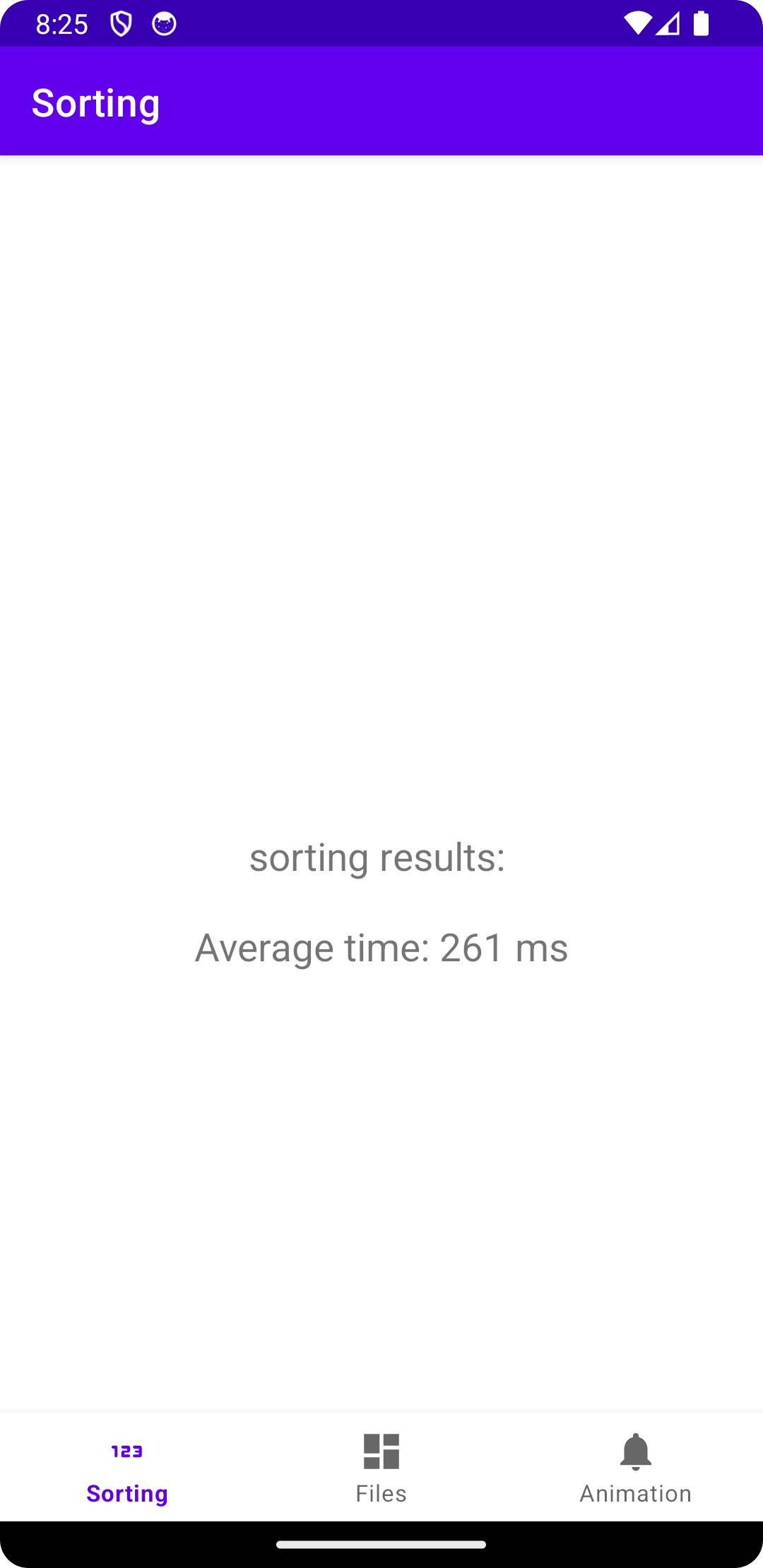
We'll show and test our app on an Android phone, and it's made for Android devices. We created the app in Android Studio using Java and XML. To test how good the phone is, we'll do three things:

* For the CPU (how fast the phone thinks), we'll see how long it takes to sort a list five times and take the average time.
* For the GPU (how well it shows graphics), we'll use an animation made with the OpenGL library.
* For memory (how it stores and uses data), we'll see how fast it can move a big list of 10,000 numbers.

**2. Analysis**

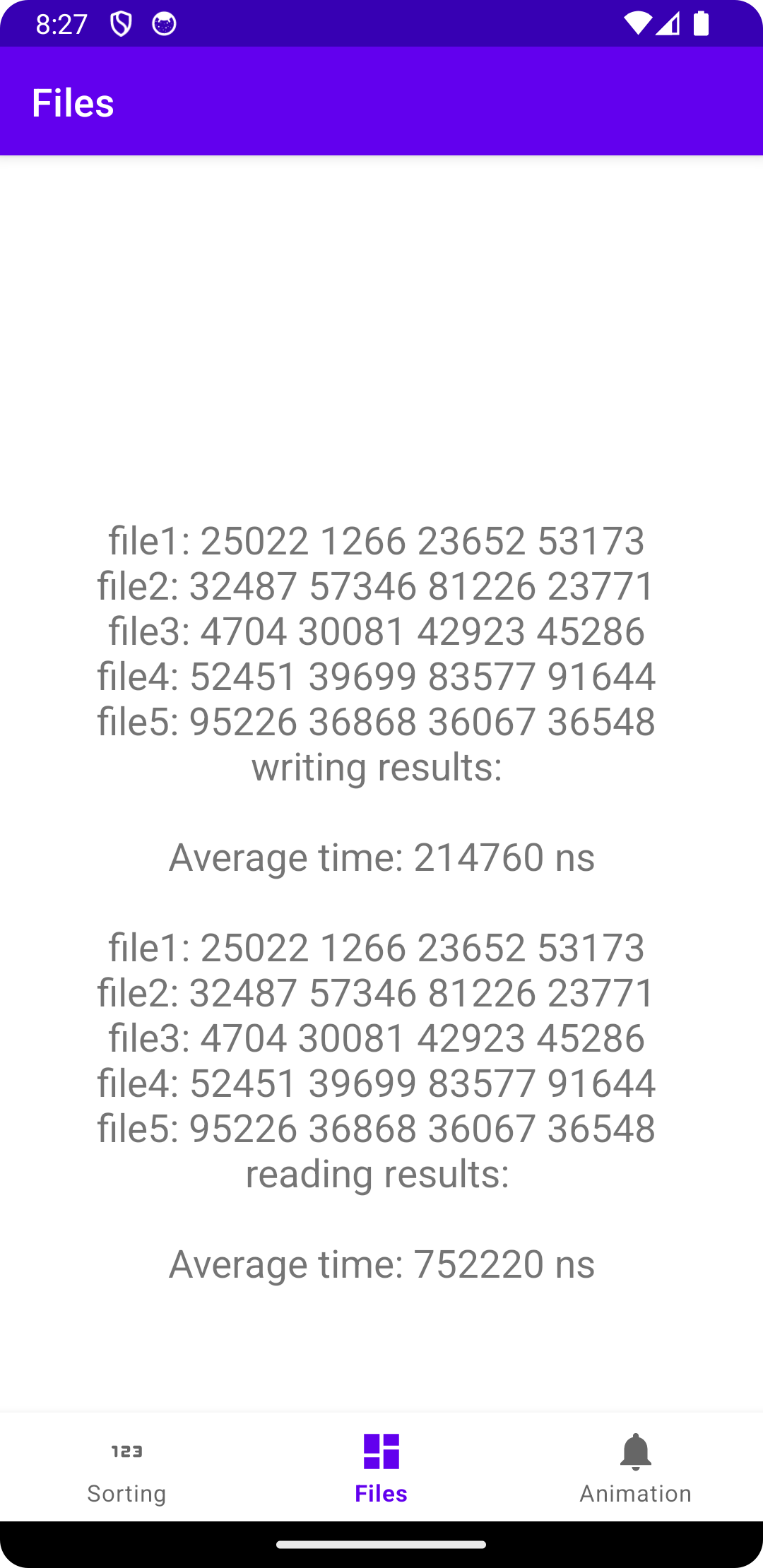
2.1. Testing the CPU performance

For testing the CPU's performance, the approach involves sorting a large array of random numbers. This process is repeated five times to get an average measure of the CPU's efficiency. The sorting task is intensive and challenges the CPU, providing a realistic assessment of its processing power. The use of a large array ensures a comprehensive evaluation of how well the CPU can handle demanding computational tasks.



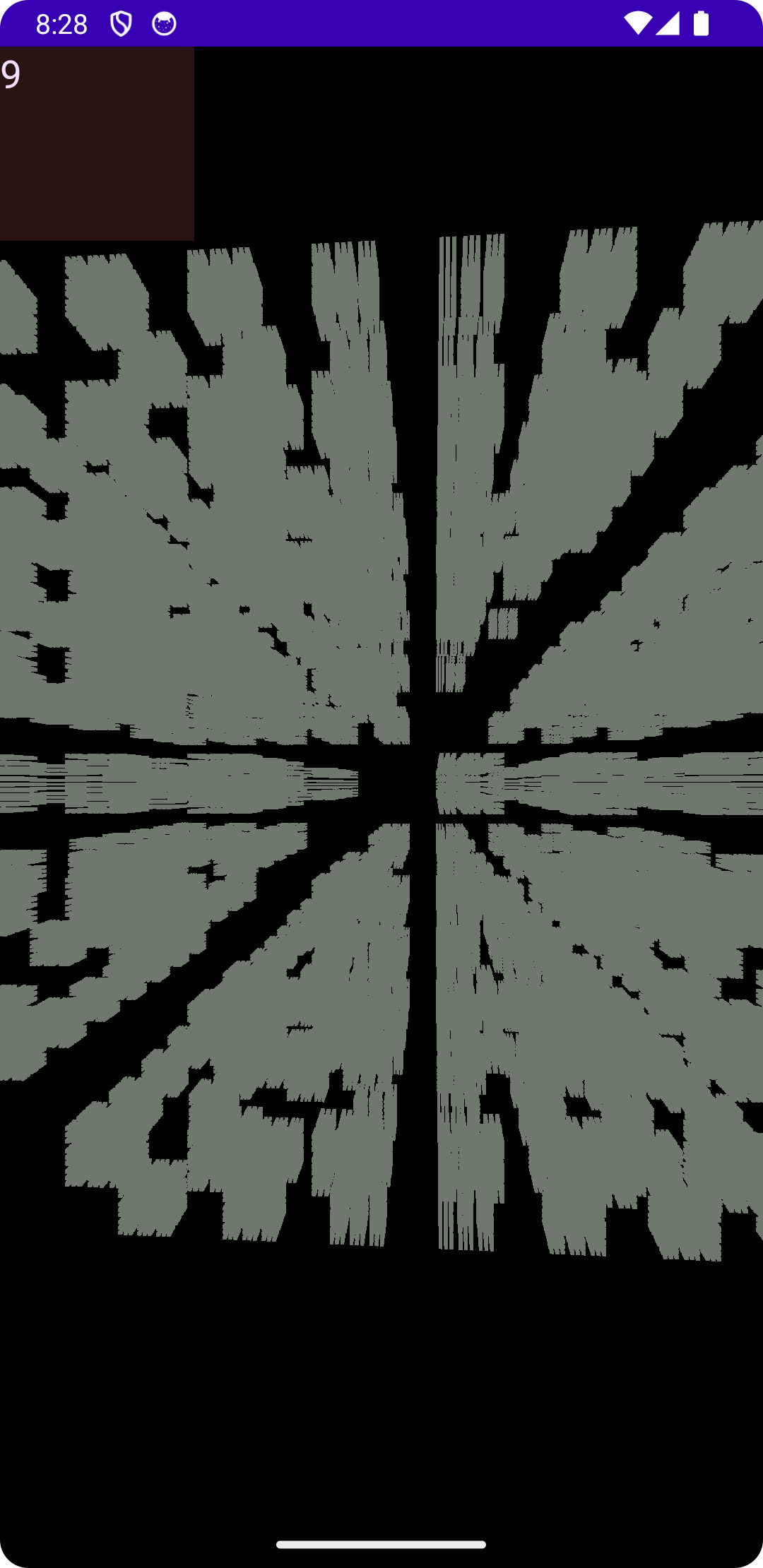
2.2 Testing Memory Performance

The memory performance is gauged by measuring the time taken to transfer a file containing an array of 10,000 integers. This process evaluates both the read and write capabilities of the phone's memory. By moving a substantial amount of data, the test provides insight into the memory's efficiency in handling large-scale data operations, a critical factor in the overall performance of the mobile device. This practical approach offers a realistic view of how the memory performs under typical usage scenarios.



2.2 Testing GPU Performance

To assess the GPU performance, the application utilizes an animation created using the OpenGL library. This method effectively measures the GPU's capability to handle graphic-intensive operations. The animation stresses the GPU, giving a clear picture of its performance in rendering complex graphics, which is a common requirement in modern mobile applications.



**3. Design**

For the application design, I used Android Studio, because it's well-suited for Android app development. The application is coded in Java, which handles the app's operations, and XML, which is used for designing the user interface.

The app's interface is straightforward and functional, with separate sections for CPU, memory, and GPU performance tests, as shown in the screenshots. The layout is uncomplicated, aiming to present the benchmarking results without unnecessary complexity.

Each test's results are displayed on its respective screen: the CPU test shows the average sorting time, the memory test shows file read and write times, and the GPU test displays an animation. This design ensures that the results are the focus of the user experience.

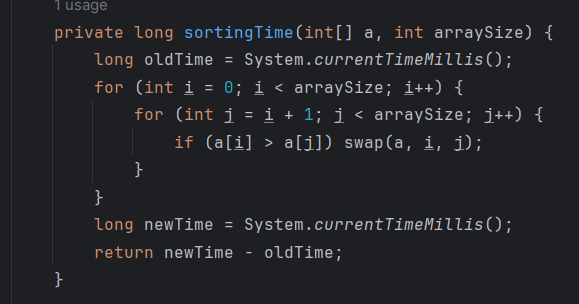
In summary, the app's design prioritizes clarity and ease of use, allowing users to understand their device's performance through direct and simple result presentations.

**4. Implementation**

4.1 Processor test

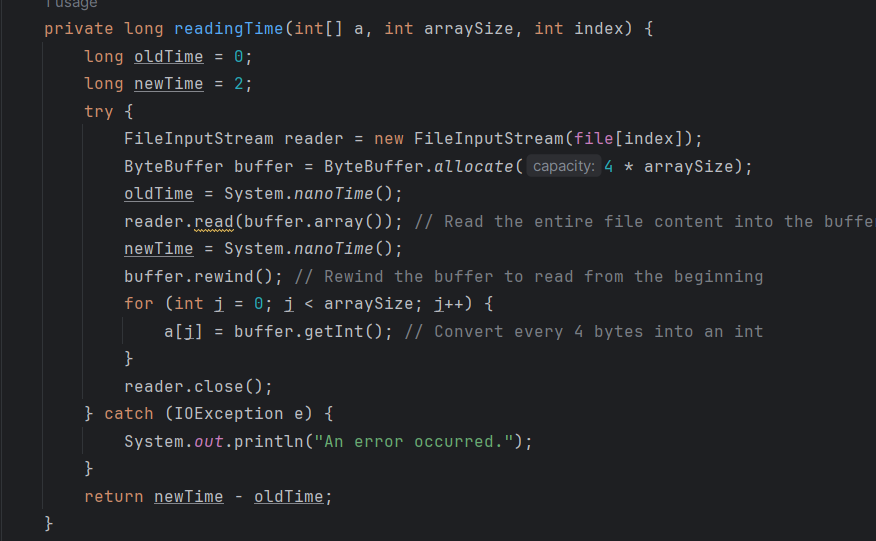
To benchmark the processor, we implemented a bubble sort algorithm on an array of integers. This method was chosen for its straightforward logic that places a consistent load on the CPU. The code performs the sorting operation on an array with a set number of elements, timing how long the CPU takes to complete the task.

Here's the method for the CPU test:

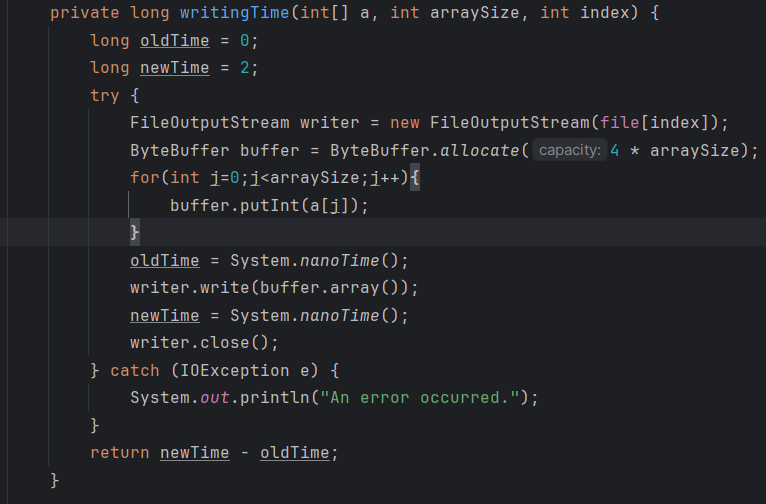


4.2 Memory Test:

For evaluating memory performance, the test was designed to read data from a file and time the operation. This test directly assesses the read performance of the device's memory. An array of integers is read from a file, capturing the time taken to complete the process using system nanotime for higher precision.

Here's the method for the reading test:

Here’s the method for writing test:



4.3 GPU Test:

The GPU test is designed to measure the rendering capabilities of the graphics processing unit by drawing a 3D object. This object, a simple farmhouse, is rendered using OpenGL ES 2.0 within the app.

The Farmhouse class parses 3D model data into vertex and face lists, converts these into buffers, and then uses shaders to control how the model is drawn to the screen. The vertex shader manages the position of the vertices, while the fragment shader handles the color.

For the actual drawing, the Farmhouse class's draw method is called, which utilizes OpenGL functions to set the program and draw the elements based on the data in the buffers. This method is passed a transformation matrix (mvpMatrix) that determines the model's position and orientation in 3D space.

The OpenGLES20Activity and MyGLSurfaceView classes set up the environment to display this OpenGL rendering. MyGLSurfaceView creates an OpenGL ES 2.0 context and sets a custom MyGLRenderer as the renderer for the surface view. This renderer is responsible for managing what is displayed and handling the drawing cycle of the 3D object.

This test is significant because it simulates a real-world GPU task that a typical mobile game or 3D application would perform, thereby providing a practical measure of the GPU's performance.

Here's a conclusion and bibliography section tailored to your project:

**5. Conclusions**

The objective of this project was to create a mobile benchmarking application specifically for Android devices using Java and XML within the Android Studio environment. The project's tasks were completed successfully, and the application met its intended goals.

A significant challenge was the selection of appropriate technologies and programming languages. Initially, other frameworks and languages were considered, but the choice of Java and XML was finalized due to their native support and optimization for Android development, as well as the extensive resources available for these technologies.

In terms of improvements, future iterations of the application could include a more engaging user interface and a robust results comparison feature. Integrating the app with a backend service to store benchmark results in a cloud database would also be a valuable addition, allowing for performance tracking over time and across different devices.

**6. Bibliography**

* + - * + ["Create your first OpenGL App with Android Studio"](https://www.youtube.com/watch?v=yip5CpVeOPI&ab_channel=SylvainSaurel)
        + ["Introduction to OpenGL Programming for Beginners"](https://www.youtube.com/watch?v=b1EZfr3ZM6E&ab_channel=CodeMaster)
        + ["OpenGL ES | Android Developers"](https://developer.android.com/develop/ui/views/graphics/opengl)