Graphical user interface, application

Description automatically generated

Benchmark for Android Devices

Andreica Ioan-Paul | Structure of Computer Systems | T.U. Cluj-Napoca | 2021

Contents

[1. Introduction 2](#_Toc84873589)

[1.1 Context 2](#_Toc84873590)

[1.2 Specifications 2](#_Toc84873591)

[1.3 Objectives 2](#_Toc84873592)

[2. Bibliographic study 3](#_Toc84873593)

[3. Design 3](#_Toc84873594)

[3.1 User Interface 3](#_Toc84873595)

# Introduction

## Context

The goal of this project is to implement a mobile solution for fast and easy testing of core hardware features (CPU, STORAGE, GPU).

In implementing the tests, a variety of methods will be used based on the tested component.

For CPU testing we will test the processor power by using various mathematical operations meant to stress the processor. The time in which the operations complete will provide the score of the processor under test. Some operations include: calculation of Fibonacci function, calculation of PI, calculation of prime numbers, encryption techniques like SHA-1 etc.

For the GPU we will test the performance of a graphics virtualization by issuing many draw calls using various techniques learned from graphics processing, like Bezier Curves, particles, using OpenGL. The score will be calculated based on the FPS (frames-per-second) in correlation with the number of draw calls and object on screen.

For STORAGE testing we will use multiple read and write calls and compare the speed of the operations.

The application can be used by anyone who wants to measure the performance of their android phone and compare it to other devices.

## Specifications

The program will run on mobile devices supporting recent android versions. It will be written in a combination of Java and C++ using the Android Studio IDE. Simulation using android emulators on PCs will most likely be possible and allow us to test various mobile phone configurations to build a relevant database of scores that the user can later check to compare its own device.

For storing the user scores, Firebase or a similar database solution will be used.

An advanced solution would be to create a separate storage microservice using Java or .NET that will handle storing the scores from the user benchmarks in a MongoDB database, the clients will only need to make simple API calls to the service, and we can use a free solution like Heroku to host the service and ensure it is alive at any time.

## Objectives

Implement reliable testing algorithms for the mentioned hardware components, create a user-friendly interface that is straight forward and still appealing, ensure that the data generated by the tests is always saved and stored safely, since a benchmark application that uses a comparison method to give meaning to the benchmark scores is useless without a comprehensive database.

# Bibliographic study

A mobile benchmark application requires 3 major components: a good UI that makes it easy for everyone to test their devices, reliable testing algorithms, and a data store to compare the test scores.

The most difficult part will be testing the GPU since the only solutions I found are using a combination of C++ for creating and drawing the objects in the environment, and Java for hosting the environment. The challenge comes from the fact that I never worked with graphical processing on Android before, and I never integrated C++ classes in Java applications. Other than that, the solutions found are straight forward: make draw calls to the GPU, constantly measure the FPS and increase the draw calls progressively to stress the GPU. The average FPS will provide the final score.

# Design

## User Interface

The user will only need to press a button to run a full benchmark test, alternatively he can select only a component to test, since the tests don’t depend on each other and can be run indicidually. During the tests, the user will be provided with some sort of status updates, or other visual indication that the app is still running. After the tests are finished the user will see the final scores for each component and a chart rating his device for every individual component or for the overall score.

A user profile will also be created when the app is first started that will contain a device ID and the device details (manufacturer, model, configuration, hardware resources). Additionally, the user may create a profile to save his benchmark results if he ever needs them in the future, this profile require a simple email and password configuration.

# Bibliography

* “Benchmark (computing)” [Online]: <https://en.wikipedia.org/wiki/Benchmark_(computing)>
* ” How to write an Android CPU benchmark tool” [Online]: <https://www.androidauthority.com/write-an-android-cpu-benchmark-part-1-679929/>
* “CPU Benchmark : Measure the power of Android devices” [Online]: <https://hackernoon.com/cpu-benchmark-measure-the-power-of-android-devices-4aadb595cb61>
* “GPU Emulation Stress Test” [Online]: <https://github.com/google/gpu-emulation-stress-test>