App Report 4 – Part 4

Deadline	November 23th, at 11:59 PM (To be exempted of the final exam)[1] (see the end of the document) December 3rd, at 11:59 PM (Final date)
Deliverable	App Report Pdf
Possible points	**45 Individual points**

App Report – Deliverable Overview

This App Report will focus on profiling and analyzing micro-optimization strategies in a third-party Open-Source APP.

Disclaimer:

As you may know, we expect you to deliver your work with the **BEST** quality possible. You are expected to use the knowledge obtained through your entire undergrad program and apply it to everything you do in this course. What we understand for excellent quality is described as follows:

1. Clear images: **Do not** upload pixelated or unreadable images.
2. Be organized: keep the same format for all sections in your report.
3. Implementation: follow the criteria learned in class, in laboratories, and in the book. **Please read the instructions for each one of these resources carefully**.

Remember that if you have any questions about something, each TA and Professor has 2 hours of attention a week; please use them and ask if any topic is not clear.

App Report – Detailed Description

Here you can find a detailed description of what you have to deliver in part 4 of the app report.

Deliverable Description

1. Include in your report the repository link. **If it is not included, your deliverable is not valid.**
2. **[Max 15 points]** For analyzing performance, you must use the profiling tools for each platform. In order to analyze the performance correctly, you should at least make the following analysis:
   - **Define scenarios**. A scenario is a sequence of activities that the user can do in the application. For example, if the application is Chrome, a possible use case would be to open YouTube and play a video.
   - **Execute these scenarios in the respective profiler and evaluate the application's performance** in terms of the IDE statistics (CPU and GPU consumption, number of threads executed, and power consumption).
   - **For each scenario, you should analyze:**
     - **i.** *GPU rendering analysis*.
     - ***ii.*** *Overdrawing*.
     - ***iii.*** *Memory management*.
     - ***iv.*** *Threading***.**

3. **[Max 15 points]** After defining the performance scenarios and being able to use the profiling tools, you should be able to identify performance problems and strengths using the profiling tools. For each scenario, you should:

   a. *GPU rendering analysis*: you have to evaluate the graphs' rendering time and identify possible problems and strengths of the app.
   b. *Overdrawing*: Likewise, you have to do overdrawing analysis for several views of your application and identify possible problems and strengths of the app.
   c. *Memory management*: in order to identify possible problems and strengths of the app, you should at least answer this answer the following questions:
      - i. Does the app have memory leaks? Which? Where?
      - ii. How is the application's RAM consumption for the different use scenarios defined in the section?
      - iii. Research about libraries that allow you to handle leak management in the application you are studying.
      - iv. When and how frequently is the GC used? Are there deep allocations patterns or heap dumps?

v. Threading: Identify at least (i) how/where are the threads created / async feature used? (ii) possible locks on the main threads. Also, describe how the app is affected by the lack/usage of this feature. How does the multithreading/async feature affect/contribute to the performance of the application?

4. **[Max 15 points]** Identify and analyze micro-optimization
   a. Identify the micro-optimization strategies used in the app.
   b. Per each micro-optimization found answer:
      i. What is the micro-optimization? (show snippets of code and also the location in the code)
      ii. Where is it located?
      iii. Why is it considered a micro-optimization?
      iv. Identify the purpose of implementing it.
   c. Likewise, you are expected to identify parts of the code that can be optimized and describe what optimization you would do and how that would improve performance.

"Since you are using snippets of code, you should adequately describe and identify the relevant part of the code that you are using. "

- In this app report, you have to write a technical audit report, in which you identify and report technical issues (related to the course topics) in a detailed way,

- **If there is no evidence of this competence (**"In this app report you have to write a technical audit"..**) for the app report deliverable, you will get 0 points as the final grade of it.**
- Remember, you have until the deadline to send us your report in BN (No Google docs or similar will be accepted). **If you do not follow the instructions, we will grade you with a 0.**

Footnote 1:

This App-Report has a special condition for the delivery.

If you have an outstanding app report, you will be **exempted** of presenting the final exam and all the final exam points are going to be assigned to your grade. But for this you have to meet the following requirements:

- You must deliver the app report at latest one week before the final deadline. Which means that you should deliver at latest **November 23th 11:59 pm.**
- **You should include all the requirements of the app report 3.** As you know the app report is a cumulative work during all the semester
- Your app report has to be outstanding. This means that only after our grading process you will now if you are exempted or not.

Before the deadline (**November 23th 11:59 pm)** you can use the office hours to ask us about your progress and advice.

In the case that you are not exempted, you have to present the final exam and you can use the received feedback to improve your app report for the final deadline.