DV2549 — Group Project
Resource and Memory Management in Graphics

Deadline	end of LP
Submission	Source code and report (pdf) uploaded to ItsLearning
Grading	A to F

1 Introduction

In assignments 1 and 2 you have worked on runtime resource managing techniques and memory allocation tools. One of requirements that matters the most in the context of this course is the re-usability of the proposed solutions, and in this final project you will demonstrate this.

2 Details

For this project, you will implement a 2d or 3d visualisation application of your testing scenario used in the second assignment, which involves the usage of graphical assets of some form in an interactive application. You are allowed to reuse previous projects for the rendering parts, and integrate them with your work from both assignments.

If you did not use the custom allocators in any way in the second assignment you should try to integrate them. If you cannot find a reasonable usage for the custom allocators within the resource manager and your test scenario then you should use them in some other aspect of your choice in the rendering application and explain this in the report.

3 Additional work

The minimum requirement for a passing grade (E) for the project and ultimately the final course grade is the implementation mentioned in the previous section and a **report** similar to assignments 1 and 2. Following is the proposed work to improve upon this grade.

1. D grade: you will have to extend the design and implementation of your resource manager to support at least 2 package formats and 2 asset formats.

2. C grade, you must fulfill the requirements for a D grade and add at least two more features of your choice to the resource manager and or the custom allocators implementation that were not part of the initial requirements. For each feature added or supported you have to explain briefly why they are beneficial and how they are used in your scenario. Review

carefully your implementation, maybe you have already two features implemented that were not in the initial requirements.

3. B grade, you will fulfill the requirements for a C, and you will also implement a small offline processing tool that will be used to create a package containing the assets used for the visualisation demo. See the section Packaging Tool for more details.

4. For an A grade, you will fulfill the requirements for a B grade, and you will complete one of the following tasks of your choice:

   (a) Write a report in the form of a scientific paper. See the section Paper description for more details. If you write a scientific paper, you do not have to write the report required before for an E grade.

   (b) Extend the resource manager to support streaming different levels of detail for a given resource (terrain mesh, textures, etc) on demand, keeping only a small working set of levels of detail.

   (c) Extend the resource manager to support streaming audio content on demand. There is no need to support compressed media, working on chunks of wave files is enough.

   (d) Extend the resource manager to support two different replacement policies for resources (to support for requests when there is no more space in RAM), and compare the performance of the different policies. You can read about replacement policies in [3]

   (e) Implement the Buddy Allocator, and compare its performance with the OS malloc/new operations. You can re-use the scenario used to test the stack/frame allocator. A good reference on this allocator (not the only one) is from the bitsquid blog[4].

3.1 Packaging Tool

For this part you will complement the on-line resource manager with a small off-line tool that can create a custom made package of assets ready to be used by the demo application. This package has to be at least a simple concatenation of asset files, with a header that describes the content and a unique identifier for each asset contained in the package. The package format support has to be added as well to the resource manager.

The mapping from unique id's to assets can be stored in the actual package or in an additional text or binary file created when the package is built. By implementing this custom package you will demonstrate the extensibility of your initial design from assignment 2 in terms of adding a new type of container.

There are no restrictions on the programing language used to create these package files, as it is an offline tool.

3.2 Paper description

A report of 4-6 pages for the work will be presented in a paper format. The paper can be prepared in Word, Latex or any other tool, and presented in PDF format when submitting. It can have 1 or 2 columns. It must include the following sections:

Title Give your work a title.

Abstract Explain in no more than 250 words what is the paper about.

Introduction Synthesis of the paper, and a short description of the following sections.

Background Explain about existing techniques for both problems faced in the assignments, also explain briefly if necessary about different approaches to solve the problem of resource managing. Find relevant references for this section.

Problem Statement / Motivation Within the context of your proposed scenarios in assignments 1 and 2, explain why it would be beneficial to develop your own memory management techniques and resource manager to deal with such problems. What are the limitations and similarities with other approaches.

Methods / proposed solution Explain the techniques implemented, using correct vocabulary and technical terms. If a well known technique has been implemented, simply mention it and use a good reference. If a variation of a known technique has been implemented then explain the differences and benefits of the proposed solution. You can include here UML designs, use cases or any other documentation technique used to describe your work. You should expect a reader with knowledge on the topic, so do not put too many details unless needed.

Results Write the results that you have obtained from the custom allocation techniques and from the resource manager usage. These can include the results obtained in the assignments 1 and 2 as well. Here you can talk about performance results, which are the easiest to measure, but also about architectural design results, indicating certain attributes of your design from a software engineering point of view. These can include, extensibility, portability of the code, maintainability, etc.

Conclusion General conclusions about your work in both assignments, highlighting the strong and weak (if any) points in the design and implementation of the proposed solutions. Here you can also mention software engineering design principles, or any other indicator of code and design quality.

You can reuse all the information and text from the two reports presented before. But you need to ensure that the final document is easy to read and it flows well.

4 Submission and grading

This final assignment will be done by the same groups as the previous assignments, and has to be submitted in electronic form (PDF and source code) in Its Learning. For the source code, you can provide a link to a publicly accessible repository. The workload between the participants of the group should be distributed in an even way, and reported in the final document. If a paper is presented instead of a report, then when submitting in ITsLearning leave a very short note explaining the distribution of work among the participants in the group.

References

[1] Gregory, J. “Game Engine Architecture”, Chapter 6. Resources and the File System.

[2] ZZipLib, ZLib compression library.

[3] Cache Algorithms:

[4] Allocation Adventures 3: The Buddy Allocator