CMP-7009A Advanced Programming Concepts and Techniques

Project Report - 20 January 2021

Put your project title here

Group members:
Rudy Jacques Lapeer, Jacques la Ru and others

School of Computing Sciences, University of East Anglia

Abstract

An abstract is a brief summary (maximum 250 words) of your entire project. It should cover your objectives, your methodology used, how you implemented the methodology for your specific results and what your final results are, your final outcome or deliverable and conclusion. You do not cover literature reviews or background in an abstract nor should you use abbreviations or acronyms. In the remainder of the report the chapter titles are suggestions and can be changed (or you can add more chapters if you wish to do so). This template is designed to help you write a clear report but you are welcome to modify it (at your peril ...). Finally, a guideline in size is approximately 3,500 words (not including abstract, captions and references) but no real limit on figures, tables, etc.

Chapter 1

Introduction

The introduction should be brief and comprise the following:

- A brief problem statement including the background/origins of your problem or chosen topic.

- A statement how you will tackle the problem (typically using a software product).

    The next sections show an in depth MoSCoW analysis followed by the report structure.

1.1 MoSCoW

Specify the Must-Should-Could-Won't functionality of your software product. Besides the usual itemised list (or table) you may also add a text clarifying as to why certain items are more important or why certain items will not make it to the final deliverable.

1.2 Report structure

Breifly describe what you will cover in the remainder of the report, chapter by chapter.

Chapter 2

Background

This chapter covers a literature, resource and/or (software) product review. This means you will cite journal or conference papers outlining methodologies you may (or may not) use but which are definitely relevant to your particular problem. Resource and/or product information will typcially be substantiated through internet links. You may use different sections if different subareas are part of your problem statement and/or solution. Since this chapter covers the literature, you should also update the corresponding bib file referred to in the bottom of this document and here it is called References.bib. You cite references like this: Taylor et al. [1] investigated non-linear FEA on the GPU. Morton [2] developed a file sequencing method in 1966. A website on OpenCL can be found here [3]. Etc.

Chapter 3

Methodology

Describe here various methods that will be used in your project. Use different sections for distinctly different subjects and use subsections for specific details on the same subject. Only use subsubsections or paragraphs (which are not numbered) if you believe this is really necessary.

3.1 Method 1

3.1.1 Method 1 specific detail 1

In case you need maths, here is an example to write an equation:

$$\int_{\Omega_0} \delta u \frac{\partial \mathbf{P}}{\partial X} d\Omega_0 + \int_{\Omega_0} \delta u \mathbf{b} d\Omega_0 + \int_{\Omega_0} \delta u \rho_0 \ddot{\mathbf{u}} d\Omega_0 = 0 \tag{3.1}$$

And here we show how to write a matrix equation:

$$\mathbf{X} \frac{\partial N}{\partial e_c} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \end{bmatrix} \tag{3.2}$$

3.1.2 Method 1 specific detail 2

blablabla

b lablabla

3.2 Method 2

3.2.1 Method 2 specific detail 1

3.3 Etc.

3.3.1 Etc.

Chapter 4

Implementation

In this chapter you cover the actual implementation of your project. This is likely to require figures such as Fig. 4.1.
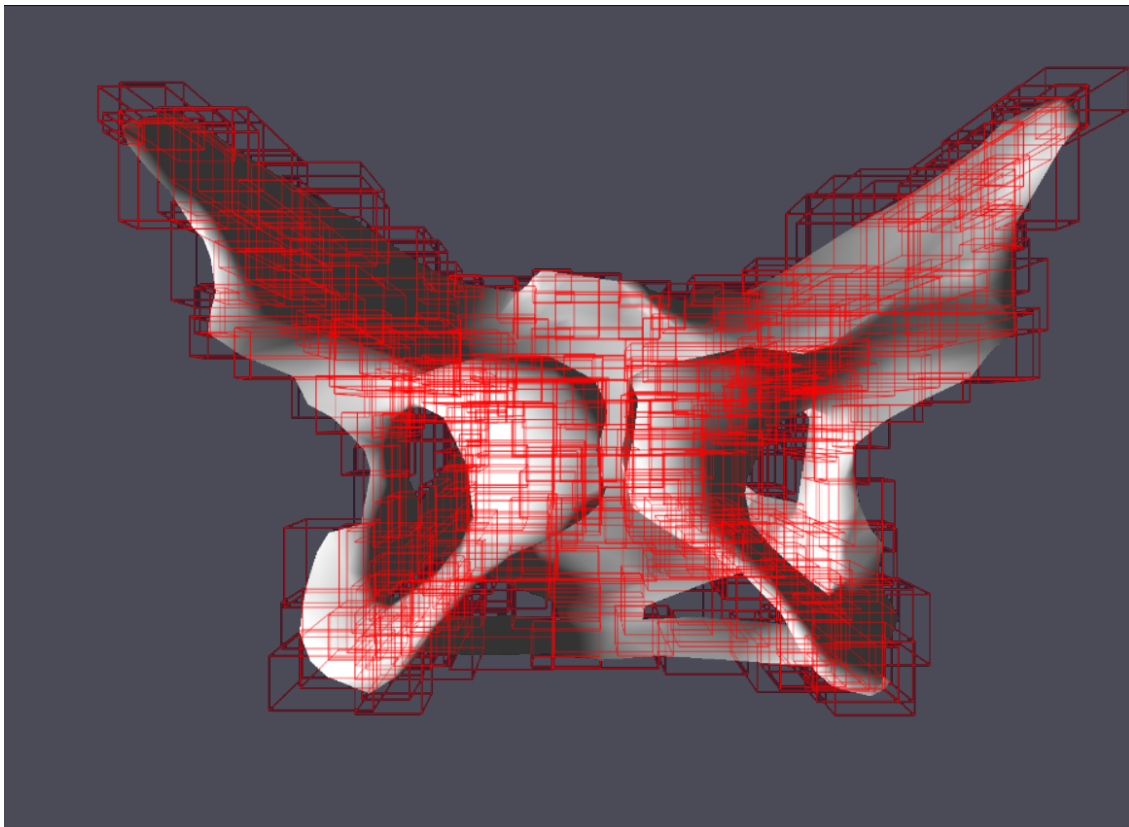

Figure 4.1: The bony pelvis model with octree based AABBs (Axis Aligned Bounding Boxes).

Or better - a UML diagram (class, sequence and state diagrams are the preferred ones) as shown in Fig. 4.2:

Or perhaps an algorithm (if it does not belong in the Methodology chapter instead).

Or perhaps a table named Table 4.1 (again could be more suitable in the Methodology chapter).

Note that code snippets or lists of crucial programming code or large UML diagrams should go in the Appendix/Appendices.
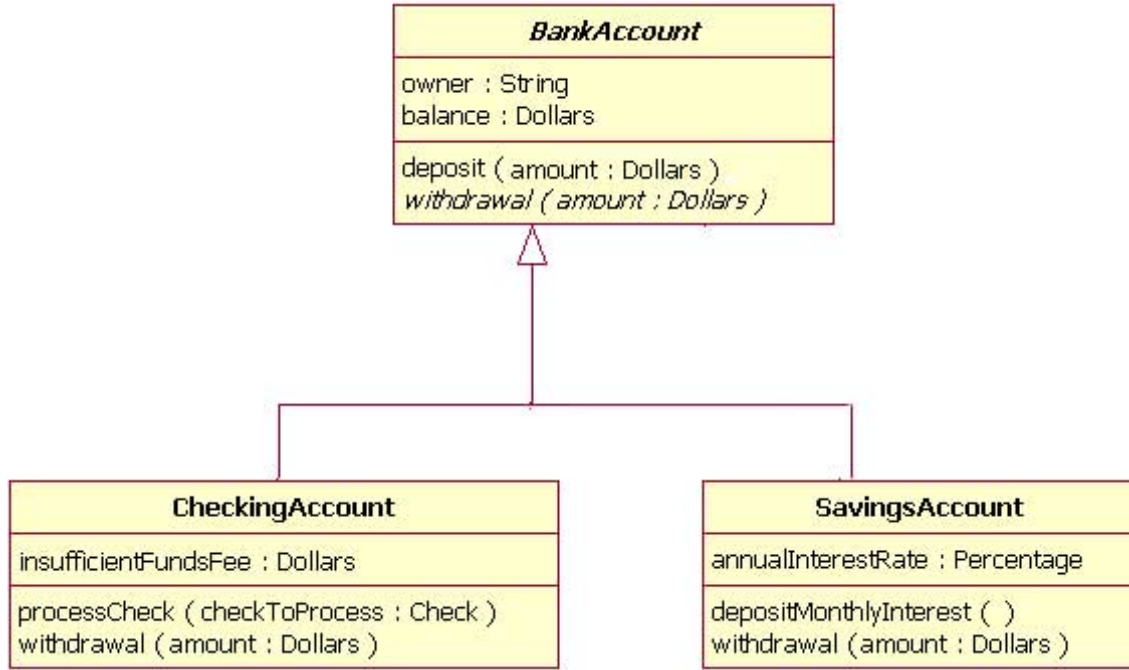
Figure 4.2: A UML class diagram.

Algorithm 1 The projection based contact method algorithm

1: Retrieve current node displacement u
   `float3 u = m_U_new[nodeIndex].xyz;`
2: Retrieve constraint plane equation
   `float4 plane = m_constraintMagnitude[nodeIndex];`
3: Calculate dot product with plane normal
   `float d = dot(u, plane.xyz);`
4: Find node penetration into the plane's negative half-space
   `float penetration = plane.w - d;`
5: **if** penetration is greater than zero **then**
6:    Find projection onto the plane surface
     `float3 proj = u + plane.xyz * penetration;`
7:    Prescribe new nodal position to be on the surface
     `m_U_new[nodeIndex] = (float4)(proj, 0.0f);`
8: **end if**

Table 4.1: Original diameters and diametral strains as reported by Sorbe and Dahlgren [4] (columns 1-2), from a previous experiment by Lapeer and Prager and reported in [5] (columns 3-4), and from the current experiment (columns 5-6).

	S-D		L-P old		L-P new	
Diameter	length	strain	length	strain	length	strain
MaVD	140.5	+1.90	129.3	+0.30	129.3	+1.43
OrOD	131.4	+0.10	-	-	119.9	+1.85
OrVD	126.9	+2.20	119.3	+0.25	119.3	+1.24
OFD	134.0	+0.40	-	-	119.7	+1.82
SOFD	-	-	-	-	113.2	-0.85
SOBD	117.1	-1.70	88.7	-1.07	88.7	-2.52
BPD	105.0	0.00	89.7	-0.21	89.7	-0.83

5

Chapter 5

Experiments (or Testing)

Describe various experiments you designed to test your software product. In case you have protocols which cover various pages, please put them in an appendix (e.g. Appendix A).

Chapter 6

Results

This chapter could be potentially merged with the previous one if testing is the only form of experiment that qualifies. Note that testing involves various levels. The ones which should definitely be done are unit tests, integration tests and user tests.

Chapter 7

Discussion

Discuss the results of your software product development. This chapter could be merged with the previous one(s).

Chapter 8

Conclusion and Future Work

Conclude your achievements and put them in perspective with the MoSCoW analysis you did early on. Be honest by declaring for example S categorised objectives which did not make it to the final deliverable rather than reversely modifying your MoSCoW in Chapter 1! Also discuss future developments and how you see the deliverable improving if more time could be spent. Do not put in subjective opinions or rants or excuses as to why something did not work out as planned in your report. A technical report is not a medium for complaints as there are other outlets for that typically well before you came to this stage (e.g. labs, e-mail etc.).

Bibliography

[1] Zeike A. Taylor, Mario Cheng, and Sébastien Ourselin. Real-Time Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2007*, pages 701–708. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] G.M. Morton. A computer oriented geodetic data base and a new technique in file sequencing. Technical Report Ottawa, Ontario, Canada, 1966.

[3] Mate Soos. AMD's OpenCL heaven and hell. https://www.msoos.org/2012/01/amds-opencl-heaven-and-hell/, 2012.

[4] B. Sorbe and S. Dahlgren. Some important factors in the molding of the fetal head during vaginal delivery - a photographic study. *International Journal of Gynecology & Obstetrics*, 21(3):205–212, 1983.

[5] R. J. Lapeer and R. W. Prager. Fetal head moulding: finite element analysis of a fetal skull subjected to uterine pressures during the first stage of labour. *Journal of Biomechanics*, 34(9):1125–1133, September 2001.

Contributions

State here the % contribution to the project of each individual member of the group and describe in brief what each member has done (if this corresponds to particular sections in the report then please specify these).

Appendix A

Put in tables of data or protocols (e.g. for testing) or code listings or UML diagrams which may take up several pages and do not sit well in the main body text.