# ASSIGNMENT 1 – OPTIMISATION TASK

Version 2.0
BSc Computing for Games
COMP350

Gareth Lewis

*"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil."*

*— Donald Knuth*

*"There is a famous rule in performance optimization called the 90/10 rule: 90% of a program's execution time is spent in only 10% of its code. The standard inference from this rule is that programmers should find that 10% of the code and optimize it, because that's the only code where improvements make a difference in the overall system performance."*

*— Richard E. Pattis*

## Introduction

In this assignment, you are required to **optimise** an **existing** project. You will document the process and the tools you have used in a blog.

Optimisation is the process of modifying a system to make it more efficient. This will often take the form of identifying via profiling and then making a change to improve performance and then re-profiling.
This assignment is formed of several parts:

(A) **Write** an initial blog which contains the following
  (i) **Describe** the project;
  (ii) **Outline** the required tools;
(B) **Review** the work of your peers
(C) **Carry out** and **document**, a series of optimisation steps
(D) **Present**, as an **individual**, a practical demo of the computer program to your tutor that will:
  (i) **Demonstrate** your academic integrity;
  (ii) **Demonstrate** your **individual** programming knowledge and communication skills.

## Assignment Setup

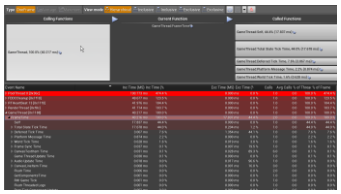
This assignment is a **programming** task. Fork the GitHub repository at the following URL:

`https://github.com/Falmouth-Games-Academy/comp350-optimisation`

Use the existing directory structure and, as required, extend this structure with sub-directories. Ensure that you maintain the readme.md file.
Modify the .gitignore to the defaults for platform or engine of your project. Please, also ensure that you add editor-specific files and folders to `.gitignore`.
For the blog, you need to log onto the following

`https://journal.falmouth.ac.uk`

You should be already attached to a 'class' and Blog page has already been created for you.


Unreal Engine CPU Profiler

cont…

## Part A

Part A consists of a **single formative submission**. This work is **individual** and will be assessed on a **threshold** basis. Answer the following questions to pass:

- What are the details of the project you are optimising?
- What tools are you going to use?

To complete Part A, prepare a blog post before **5pm on Friday of Week 2**.

## Part B

Part B consists of a **single formative submission**. This work is **individual** and will be assessed on a **threshold** basis. You will upload your coursework repository for peer review.

You will be assessed on the following criteria

(a) Submission is timely;
(b) Enough work is available to conduct a meaningful review;
(c) A broadly appropriate review of a peer's work is submitted.

To complete Part B, prepare draft versions of the computer programs. Ensure that the source code and related assets are pushed to GitHub and a pull request is made prior to the scheduled peer-review session. Then, attend the scheduled peer-review session.

You will receive immediate **informal feedback** from your **peers**.

## Part C

Part C is a **single formative submission**. This work is **individual** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) Submission is timely;
(b) Enough work is available to conduct a meaningful review;
(c) A broadly appropriate review of a peer's work is submitted.

To complete Part C, prepare draft versions of the computer programs. Ensure that the source code and related assets are pushed to GitHub and a pull request is made prior to the scheduled sprint review sessions. Then, attend the scheduled sprint review sessions.

## Part D

Part D is a **single summative submission**. This work is **individual** and will be assessed on a criterion-referenced basis. Please refer to the marking rubric at the end of this document for further detail.

To complete Part D, revise your program based on the feedback you have received. Then, upload a .zip file to LearningSpace containing the following:

(a) The source code for your project;
(b) A `readme.md` which has URL to your Dev Blog

Please note, the LearningSpace will only accept a single .zip file. You will receive **formal feedback** three weeks after the final submission deadline.

cont…

## Additional Guidance

As always, avoid underestimating the effort required to implement even simple software; always consider scope. From the proposal stage, you should consider very carefully what is feasible. The important aspect about this coursework is the profiling process; you should approach this like an experiment and document each step and iteration in the process.

A common pitfall is poor planning or time management. Many underestimate the work involved in designing and implementing multiplayer games. It simply cannot be crammed into a last minute deluge just before a deadline. There is a critical and time-consuming phase of testing! It is, therefore, very important that you begin work early and sustain a consistent pace: little and often.

## FAQ

- **What is the deadline for this assignment?**
  Falmouth University policy states that deadlines must only be specified on the MyFalmouthsystem.

- **WhatshouldIdotoseekhelp?**
  You can email your tutor for informal clarifications. For informal feedback, make a pull request on GitHub.

- **Is this a mistake?**
  If you have discovered an issue with the brief itself, the source files are available at:

  https://github.com/Falmouth-Games-Academy/bsc-assignment-briefs.

  Please raise an issue and comment accordingly.

## Additional Resources

- http://gameprogrammingpatterns.com/optimization-patterns.html
  https://docs.unrealengine.com/latest/INT/Engine/Performance
- https://docs.unity3d.com/Manual/Profiler.html
- https://msdn.microsoft.com/en-us/library/ms182372.aspx

cont…

# Marking Rubric

| Criterion | Weight | Refer for Resubmission | Basic Proficiency | Novice Competency | Novice Proficiency | Professional Competency | Professional Proficiency |
|---|---|---|---|---|---|---|---|
| Basic Competency Threshold | 40% | At least one part is missing or is unsatisfactory. | Submission is timely.<br>Considerable engagement with version control, commensurate with at least three or more commits per week.<br>Enough work is available to hold a meaningful discussion.<br>Clear evidence of programming knowledge and communication skills.<br>Adequate participation in-class peer-review activities at least at the level of basic competency.<br>No breaches of academic integrity. | | | | |
| Appropriateness of Project Description | 10% | The description lack detail or is unclear | The description is poor and requires more detail<br>There is no list of tools to be used during the optimisation process | The description is of a good level of detail but it is over-scoped<br>Some tools were listed but no detail was given on these | The description is of a good level of detail and is of appropriate scope<br>Some tools were listed and there is a brief description of the tool | The description is of a very good level of detail and is of appropriate scope<br>Some tools were listed and there is a description of good level of detail | The description is of an excellent level of detail and is of appropriate scope<br>Some tools were listed and there is a description of an excellent level detail |
| Sophistication | 10% | Little insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is poor or non-existent. | Some insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is adequate. | Much insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is appropriate. | Considerable insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is effective. There is high cohesion and low coupling. | Significant insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is very effective. There is high cohesion and low coupling. | Extensive insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is extremely effective. There is very high cohesion and very low coupling. |
| Maintainability | 15% | The code is only sporadically commented, if at all, or comments are unclear.<br>Few identifier names are clear or inappropriate.<br>Code formatting hinders readability. | The code is well commented.<br>Some identifier names are descriptive and appropriate.<br>An attempt has been made to adhere to a consistent formatting style.<br>There is little obvious duplication of code or of literal values. | The code is reasonably well commented.<br>Most identifier names are descriptive and appropriate.<br>Most code adheres to a sensible formatting style.<br>There is almost no obvious duplication of code or of literal values. | The code is reasonably well commented, with appropriate Doxygen-compatiable documentation.<br>Almost all identifier names are descriptive and appropriate.<br>Almost all code adheres to a sensible formatting style.<br>There is no obvious duplication of code or of literal values. Some literal values can be easily “tinkered”. | The code is very well commented, with comprehensive appropriate Doxygen-compatiable documentation.<br>All identifier names are descriptive and appropriate.<br>All code adheres to a sensible formatting style.<br>There is no obvious duplication of code or of literal values. Most literal values are, where appropriate, easily “tinkered” outside of the source. | The code is commented extremely well, with comprehensive appropriate Doxygen-compatiable documentation.<br>All identifier names are descriptive and appropriate.<br>All code adheres to a sensible formatting style.<br>There is no duplication of code or of literal values. Nearly all literal values are, where appropriate, easily “tinkered” outside of the source. |
| Peer Review | 10% | No peer-review submitted. | There is evidence of some engagement with peers (e.g. code review).<br>Comments to peers are somewhat constructive and provide some insight. | There is evidence of some engagement with peers (e.g. code review).<br>Comments to peers are somewhat constructive and provide much insight. | There is evidence of much engagement with peers (e.g. code review).<br>Comments to peers are reasonably constructive and provide much insight. | There is evidence of much engagement with peers (e.g. code review).<br>Comments to peers are reasonably constructive and provide considerable insight. | There is evidence of considerable engagement with peers (e.g. code review).<br>Comments to peers are highly constructive and provide significant insight. |
| Optimisation Process | 15% | Optimisation has been carried out but there is no supporting documentation | Optimisation has been carried out with some basic supporting documentation<br>The documentation lacks detail and clarity, there is also a lack of figures, charts and tables | Optimisation has been carried out with more detailed supporting documentation<br>The documentation lacks detail and clarity, but there is evidence of more analytical approach (including diagrams) | Optimisation has been carried out and the supporting documentation is of a good level<br>The documentation is of a good level of detail and there is evidence of more analytical approach (including diagrams) | Optimisation has been carried out and the supporting documentation is of a very good level<br>The documentation is of a good level of detail and the commentary on the optimisation process is of a good level of detail | Optimisation has been carried out and the supporting documentation is of a excellent level<br>The documentation is of a good level of detail and the commentary on the optimisation process is of a excellent level of detail |