WORKSHEET TASKS

Version 1.1
Computing
COMP270

Dr Ed Powley, Kate Bergel

Introduction

In this assignment, you are required to complete a number of programming tasks to accompany your studies of mathematics for 3D worlds and simulations. This assignment is divided into a series of **FOUR worksheets**, each of which requires you to **implement** mathematical methods and algorithms to solve specified problems in areas such as geometry, physics and lighting.

"Mathematics obviously has a profound influence on computer science; nearly every branch of mathematical knowledge has been brought to bear somewhere. I recently worked on a problem dealing with discrete objects called 'binary trees', which arise frequently in computer representations of things, and the solution to the problem involved the complex gamma function times the square of Reimann's zeta function. Thus the results of classical mathematics often turn out to be useful in rather amazing places."

— Donald E. Knuth, Selected Papers on Computer Science

The worlds of mathematics and programming have much in common, with rational thinking for problem solving being a key skill in both, however the translation of mathematical ideas into code is not always direct or intuitive. As with all aspects of programming, becoming familiar with (sometimes obscure) mathematical notation and developing fluency in implementing it takes time and a sustained effort; for this reason, you will work consistently across the semester by completing a series of bite-sized worksheets.

This assignment is formed of **FOUR** worksheets, numbered 1 to 4. These focus on the following topics:

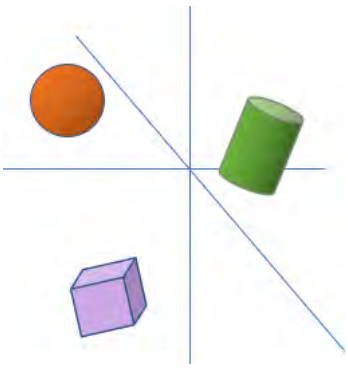
1) Parametric equations and 2D geometry;
2) Newtonian mechanics and numerical control;
3) 3D geometry and raycasting;
4) Introduction to the graphics pipeline and VFX techniques.

For each part you must:

(i) **Read** the instructions in the worksheet;
(ii) **Fork** a BitBucket repository containing a skeleton implementation;
(iii) **Complete** the implementation as specified in the worksheet;
(iv) **Submit** your solution via BitBucket pull request by the deadline specified when the worksheet is released (usually the Friday two weeks after the material becomes available).

Assignment Setup

This assignment consists of **four formative submissions**, followed by a **single summative submission**. You will receive **feedback** after each formative submission.



At the end of the study block you will be required make a final summative submission of all four of your worksheet solutions. Prepare a **single .zip file** containing your four submissions **in four separate folders**, and upload it to the appropriate submission area on LearningSpace. This submission is for archival purposes only; at this stage your work has already been marked and you have received feedback, and you should **not** submit any new, unmarked work via LearningSpace unless you have been granted permission to do so by the tutor.

This final submission is subject to the usual university policies regarding late submission or non-submission, as detailed in the course handbook — even if you have met all the formative deadlines for individual worksheets, failure to make a submission via LearningSpace by the summative deadline will be subject to penalties.

Additional Guidance

Make a submission on time and you will get a mark of at least 30% on that worksheet, even if your solution is incorrect or incomplete. A solution meeting all of the correctness and/or functionality criteria on the worksheet is required to demonstrate basic proficiency, with higher grades contingent on your solution being of a high quality. The individual worksheets give more guidance as to what constitutes "quality" for that particular exercise.

It is very important to keep up with the worksheets. Missing a deadline results in an automatic mark of 0% for that worksheet. The underlying skills being developed are also critically important to your progression as a programmer, so do not neglect the work. Do not underestimate the time it takes to complete tasks that may appear trivial when you first see them. Do not leave work until the last minute! With programming in particular, trying to "cram" the work just before the deadline is a sure path to failure. Aim for consistent, steady progress over the course of the semester.

A key skill of software development is the ability to read and follow instructions. Make sure to read the worksheet carefully to ensure that you are meeting all of the requirements — a surprising number of students needlessly lose marks by misreading the worksheet.

Nobody learns in a vacuum: you are allowed, and indeed encouraged, to discuss your work with your peers. However you must be very careful to avoid falling into **academic misconduct**, in particular **plagiarism**. If any part of your solution is **not your own individual work**, you must make this as clear as possible in your submission, for example in source code comments.

FAQ

- **What is the deadline for this assignment?**
  Each worksheet has its own formative deadline, specified on that worksheet and also communicated in class. Falmouth University policy states that summative deadlines must only be specified on the MyFalmouth system.
- **What should I do to seek help?**
  You can email your tutor for informal clarifications. For informal feedback, make a pull request on BitBucket.
- **How will I receive feedback on my work?**
  You will be given verbal feedback on your work during the session in which it is marked. If you require more in-depth feedback or discussion, please book an appointment with your tutor.
- **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 make a pull request and comment accordingly.

Additional Resources

- F. Dunn and I. Parberry (2011). 3D Math Primer for Graphics and Game Development. CRC Press.
- E. Lengyel (2011). Foundations of Game Engine Development, Volume 1: Mathematics. Terathon Software LLC.
- T. Jenkyns and B. Stephenson (2012). Fundamentals of Discrete Math for Computer Science: A Problem-Solving Primer. Springer.

See also individual worksheets.

Worksheet Rubrics and Weightings

Each worksheet is marked according to its own rubric. Please see individual worksheets for details. Each of the four worksheets is **weighted equally**, and so is worth **25%** of the overall marks for this assignment.