

Dr Ed Powley

## Introduction

*"No matter the language, learning to program will take a very long time, and is often very frustrating. I'm sorry. There is no solution. Just practice. Regular practice.*

*"That's why I always call programming a craft. You spend your life honing your craft, not a weekend.*

*"You WILL get stuck. You WILL get very frustrated. You WILL want to quit. You WILL think this is all pointless and dumb. You WILL look at the things others are doing and think 'How the hell am I going to do this?'*

*"Programming can make you feel empowered. Programming can make you feel excited. Programming can be a major source of inspiration.*

*"But it has to come from you. YOU are the driving force here. I and others can only point you in a direction."*

— Ólafur Waage

In this assignment, you are required to complete a number of tasks to accompany your studies of the principles of computing. This assignment is divided into a series of bite-sized **worksheets**. The worksheets require you to **design**, **annotate**, and **write** a series of computer programs according to instructions, as well as to **apply** mathematical techniques to **solve** problems.

In order for programmers to communicate with each other regarding the technical aspects of a game development project, they must have good computational thinking skills, a strong foundational knowledge of computing principles, applied knowledge of program design notations and annotations, and a working knowledge of particular programming constructs (often as a result of writing their own versions). Such knowledge and skills take time and a sustained effort to develop. For this reason, you will work consistently across the semester by completing a series of bite-sized worksheets.

This assignment is formed of **NINE** parts: Worksheets 1 to 9. These address the following learning objectives:

- (1) **Apply** basic computational thinking to puzzle solving;
- (2) **Use** number representation systems including binary and hexadecimal;
- (3) **Express** simple algorithms using notations such as flowcharts and pseudocode;
- (4) **Reason** using the operations of Boolean logic;
- (5) **Utilise** theoretical concepts such as computational complexity;
- (6) **Write** programs using algorithm strategies based on recursion;
- (7) **Implement** programs making use of advanced data structures;
- (8) **Calculate** geometric quantities using vectors;
- (9) **Understand** how computer processors execute low-level machine code.

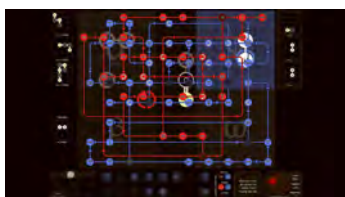
For each part you must:

- (i) **Read** the instructions in the worksheet;
- (ii) **Complete** all of the problems presented;
- (iii) **Submit** your solution according to the instructions on the worksheet, and by the deadline specified on the worksheet.

## Assignment Setup

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

Each worksheet contains detailed submission instructions. You will generally be required to either **fork** a repository on GitHub and submit a **pull request** containing your solution, or to complete a set of mathematical problems using pen-and-paper and hand this in physically in class.



SpaceChem is a puzzle game in which players must apply computational thinking to build circuits which assemble chemical molecules.

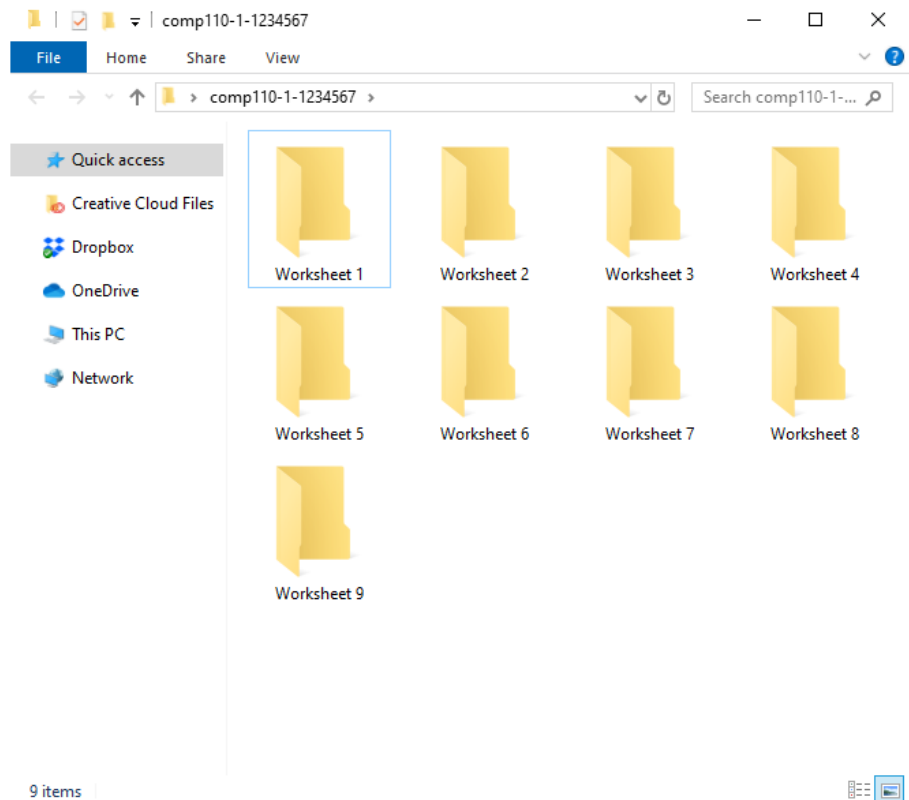


Figure 1: Recommended folder structure for your final summative submission.

At the end of the semester you will be required make a final summative submission of all nine of your worksheet solutions. Prepare a **single .zip file** containing your submissions **in separate folders**, and upload it to the appropriate submission area on LearningSpace. The folder structure within your zip file should resemble that shown in Figure 1. 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 at least a 30% mark on that worksheet, even if your solution is incorrect or incomplete. This is just below the pass threshold of 40%, so even an incomplete solution submitted on time is often enough to pass. 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, but bear in mind that a major purpose of these worksheets is to assess your ability to communicate complex computational ideas in English, in notation and in program code. Thus pay particular attention to the precision and clarity of your written communication, and the readability and maintainability of your source code.

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 GitHub.
- **How will I receive feedback on my work?**  
You will receive verbal feedback during the timetabled worksheet support and sign-off sessions. You will also receive formative feedback on some worksheets through automated marking, and on others through written comments on e.g. GitHub pull requests. You will receive formal feedback three weeks after the summative deadline.
- **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 nine worksheets is **weighted equally**, and so is worth 11.1% of the overall marks for this assignment.