

Dr Ed Powley

Introduction

In this assignment, you are required to **design**, **annotate**, and **write** a series of computer programs according to instructions.

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 **five** parts. Each of parts A–E corresponds to one worksheet, addressing the following five learning objectives:

- (A) **Apply** basic computational thinking to puzzle solving;
- (B) **Express** simple algorithms using notations such as flowcharts and pseudocode;
- (C) **Utilise** theoretical concepts such as computational complexity;
- (D) **Implement** programs making use of advanced data structures;
- (E) **Understand** how computer processors execute low-level machine code programs.

For each part you must:

- (i) **Read** the instructions in the worksheet;
- (ii) **Complete** all of the problems presented in the worksheet;
- (iii) **Submit** your solution via GitHub pull request, by the deadline specified on the worksheet;
- (iv) **Complete** an online quiz, scoring at least 90%. Note that you may attempt the online quiz as many times as you wish, without penalty.

Assignment Setup

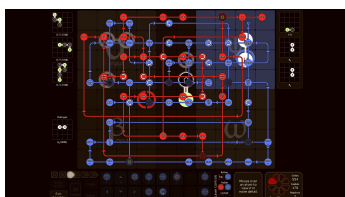
This assignment consists of **five formative submissions**, followed by a **single summative submission**. You will receive **feedback** after each formative submission.

Each worksheet contains detailed submission instructions; you will generally be required to **fork** a repository on GitHub and submit a **pull request** containing your solution.

You will also be required to complete an online quiz, which can be found in the appropriate area of LearningSpace. These quizzes give you an opportunity to develop the mathematical skills that are essential for professional game development. You must achieve a score of at least 80% on each quiz; however you may re-take each quiz as many times as you wish.

"Bad programming is easy. (People) can learn it in 21 days, even if they are dummies...(Good programming requires a) willingness to devote a large portion of one's life to deliberative practice...So go ahead, buy that book; you'll probably get some use out of it. But you won't change your life or your real expertise as a programmer in 21 days...How about working hard to continually improve over 24 months? Well, now you're starting to get somewhere..."

— Peter Norvig



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

At the end of the semester you will be required make a final summative submission of all five of your worksheet solutions. Prepare a **single .zip file** containing your five worksheet submissions **in five 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 basic pass 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, 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 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.

Marking Rubric

Criterion	Weight	Refer for Resubmission	Basic Competency	Basic Proficiency	Novice Competency	Novice Proficiency	Professional Competency
Basic competency threshold	40%	A reasonable attempt at one or more worksheets was not submitted by the corresponding formative deadlines. At least one of the online quizzes was not passed.	A reasonable attempt at each worksheet was submitted by the corresponding formative deadlines. Each of the online quizzes was passed, with either a score of 80% by the quiz’s formative deadline or a score of 100% after the deadline. There is no evidence of academic misconduct.				
Worksheet A	12%	Please see worksheet for marking criteria.					
Worksheet B	12%	Please see worksheet for marking criteria.					
Worksheet C	12%	Please see worksheet for marking criteria.					
Worksheet D	12%	Please see worksheet for marking criteria.					
Worksheet E	12%	Please see worksheet for marking criteria.					