

Brian McDonald

Introduction

In this assignment, you are required to **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 **three** parts. Each part corresponds to one worksheet, addressing the following learning objectives:

- (A) **Apply** the basic principles of programming in C++;
- (B) **Implement** algorithms in C++ and SDL2 using notations such as flowcharts and pseudocode;
- (C) **Apply** theoretical concepts such as communication protocols and object-orientated design patterns.

For each worksheet you must:

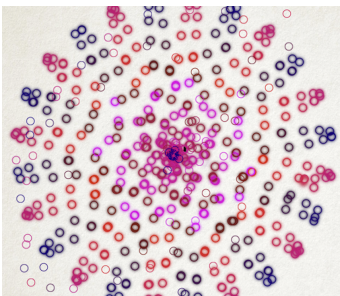
- (i) **Read** the instructions in the worksheet;
- (ii) **Complete** all of the problems presented in the worksheet;
- (iii) **Make** the solution available to the module leader for formative review;
- (iv) and **submit** the solution summatively on the LearningSpace.

"As soon as we started programming, we found out to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs."

— Maurice Wilkes

"C++ is history repeated as tragedy. Java is history repeated as farce."

— Scott McKay



The **MetaMakers** undertake cutting edge Computational Creativity research at **Falmouth University**. Their main is to develop, combine, and extend artificial intelligence techniques to build inspirational software, and explore the philosophy of what it means for software to be creative.

Assignment Setup

This assignment consists of **multiple formative submissions**, followed by a **single summative submission**. You will receive **immediate informal 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.

At the end of the semester you will be required make a final summative submission of all of your worksheet solutions. Prepare a **single .zip file** containing your five worksheet submissions **in separate folders**, and upload it to the appropriate submission area on LearningSpace. This final submission is subject to the usual university policies regarding late submission or non-submission—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 late submission 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.

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

Please see individual worksheets.

Marking Rubric

Criterion	Weight	Refer for Resubmission	Basic Competency	Basic Proficiency	Novice Competency	Novice Proficiency	Professional Competency
Worksheets	33% × 3 worksheets	A reasonable attempt at the worksheet is not submitted by the deadline.	The submission is either late or on-time. The submission is a reasonable attempt, but is incomplete or incorrect.	The submission is on time. The submission is complete and correct. The submission has significant quality issues.	The submission is on time. The submission is complete and correct. The submission has some quality issues.	The submission is on time. The submission is complete and correct. The submission has very few quality issues.	The submission is on time. The submission is complete and correct. The submission has almost no quality issues.