ONLINE TECHNICAL PORTFOLIO

Version 2.1
BSc Computing for Games
COMP330

Gareth Lewis

Introduction

In this assignment, you will develop an online portfolio, exploring:

(i) The development of a demonstration application using JavaScript / Canvas
(ii) Creation of a technical poster to demonstrate your solution process to an you found of interest in development

"Software testing is a sport like hunting, it's bug hunting."

Amit Kalantri

'Show, don't tell' is the clarion call from many recruiters to developers looking to get into industry. As a developer, the challenges are just what and how to show. Presenting links to YouTube videos of group projects can be problematic as it can be hard for a recruiter to understand your precise role in the development project and, often, recruiters are reluctant to evaluate anything that they can't play in their hands. Providing links to project zip files and/or your games on Steam, Itch.io Ludem Dare will give recruiters access to what you have made, but there is the obvious barrier for time poor recruiters in needing to download, install and play your offering.

Conversely, developing applications in JavaScript and HTML5 gives you, as a developer, an environment where you can experiment with game engine technology, and game genres and play mechanics to create web-based applications that can be easily reached and have their content inspected by interested recruiters, making it far easier to get to a technical interview.

"Reusability is key in reducing bugs and coding quickly. The more I use a piece of code, the more confident and familiar I become with it, which in turn significantly speeds up my development time."

Robert Duchnik,
jQuery Plugin Development In 30 Minutes

Technical interviews present their own issues, as a developer with some considerable development experience from your degree studies, how do you articulate your technical designs into code, how do you demonstrate your understanding of complex design patterns and how do you show that you can make reasoned decisions based on a mixture of technical understanding and experiential heuristics. Making and presenting technical posters will massively help you to develop those vital skills.

This assignment is formed of several parts:

(A) **Jam** your game, application or technical demo
   i. Use your time before week 5 to explore JavaScript to create some interesting demo pieces and host them on your Digital Oceans server
   ii. In week 5 there will be a peer review of your jamming activities
(B) **Refine** your jam activities to create an online portfolio
   i. Your portfolio will consist of at least one JavaScript application to demonstrate your technical expertise and creative interests
   ii. The portfolio will be hosted on you Digital Ocean server and be globally available on the internet

"Java is to JavaScript what Car is to Carpet."

Chris Heilmann

cont...

"JavaScript's global scope is like a public toilet. You can't avoid going in there, but try to limit your contact with surfaces when you do."

Dmitry Baranovskiy

(C) **Create** a technical poster that:

i. Showcases an issue you ran into during developing your technical portfolio and the approach you took to solving it

(D) **Present** of your work:

i. A demonstration of your online portfolio

ii. A walk-though of your technical poster

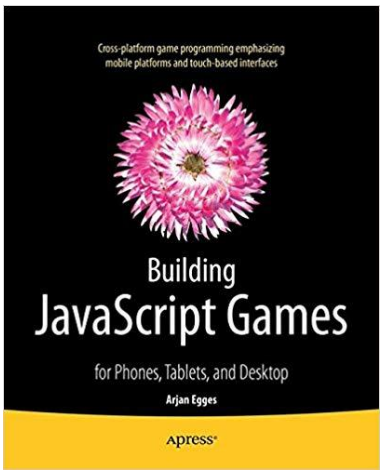
Part A

Part A 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 A**, upload your JavaScript jamming results onto your Digital Oceans server as a website and make the URL available on the scheduled Learning Space peer-review workshop. Then, attend the scheduled session.

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

Part B

Part B 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 B, upload your submission to the Learning Space. Please note the Learning Space will only accept a single .zip file. Please submit parts B & C together as a single zip file.

"You're building your own maze, in a way, and you might just get lost in it."

Marijn Haverbeke
Eloquent JavaScript: A Modern Introduction to Programming

You will receive **formal feedback** from your tutor three weeks after the final submission deadline.

Part C

Part C is a **poster**. 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 C, upload your submission to the Learning Space. Please note the Learning Space will only accept a single .zip file. Please submit parts B & C together as a single zip file.

You will receive **formal feedback** from your tutor three weeks after the final submission deadline.

Part D

Part D is an **informal presentation**. 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, attend your allotted presentation time.

You will receive **formal feedback** from your tutor three weeks after the final submission deadline.

cont...

Additional Guidance

A common pitfall is poor planning or time management. Often, students underestimate how much work is involved in first learning programming concepts and then actually applying them. Programming is quite unlike other subjects in that it cannot be crammed into a last minute deluge just before a deadline. It is, therefore, very important that you begin work early and sustain a consistent pace: little and often. The live deployment, in this assignment, is an added dimension. Aim to have your portfolio running on your Digital Ocean server as soon as possible and frequently update it as your develop the functionality of your portfolio.

Likewise, the technical poster requires some significant considerations in the choice of an issue that is small enough to be covered in a poster but big enough to contain enough depth and interest as to not make the poster incredibly superficial, vague or confusing.

Be sure to give some careful consideration to both the nature and the scope of the portfolio work you are going to undertake. It is well worth creating something that is small and well-defined that you can actually use to help you in your post-graduation recruitment process rather than making a huge, but unfinished, application or an application you have no real interest in.

Finally, it's worth thinking about if and how JavaScript in general and your portfolio piece, in particular, could be leveraged to help with your group and/or research projects.

FAQ

- **What is the deadline for this assignment?**
  Falmouth University policy states that 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.

- **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

- https://www.w3schools.com/js/
- https://www.w3schools.com/graphics/canvas_intro.asp
- https://webglfundamentals.org/
- https://www.youtube.com/watch?v=Fy0aCDmgnxg
- https://www.youtube.com/watch?v=HQYsFshbkYw
- https://www.youtube.com/watch?v=Xq3isov6mZ8
- https://www.youtube.com/channel/UC-yuWVUplUJZvieEligKBkA
- https://www.youtube.com/channel/UCfVFSjHQ57zyxajhhRc7i0g
- https://www.youtube.com/watch?v=ybLZyY655iY
- https://github.com/ocornut/imgui/tree/master/examples

cont...

Marking Descriptors: Online Technical Portfolio

Criterion	Weight	Refer for Resubmission	Novice Competency	Novice Proficiency	Professional Competency	Professional Proficiency	Expert Competency
Threshold	40%	Parts A, B, C or D are not completed or are unsatisfactory	Parts A, B, C & D are complete and submission is timely. Enough work is available to hold a meaningful discussion. Provided a meaningful review of a peer's work. Clear evidence of programming knowledge and communication skills. Attend & deliver presentation No breaches of academic integrity				
Software Engineering	20%	No code submitted	Code structure and function is confusing and hard to follow Few or no comments Project structure is ad hoc	Code structure and function makes some sense, but has evidence of 'left over' classes, meaningless functions and a sense of 'code & fix' rather than planned development Some evidence of comments Some evidence of planning to project structure	Class hierarchy makes sense. Code functionality makes reasonable sense and little evidence of 'code & fix' Fairly well-commented codebase Project structure shows clear organisation without lots of unused assets	Class hierarchy describes problem domain reasonably well Code functionality makes sense but could benefit from refactoring / simplification Well commented codebase Project structure makes sense with assets in meaningful folders	Class hierarchy describes problem domain very well Code functionality makes sense and is straight forward Well commented codebase with evidence of self-commenting code Project structure makes sense with assets in meaningful folders
Poster Content	10%	There is no poster or it does not describe problem solving and engineering of solutions	Problem is trivial and/or poorly defined The engineering of the software (e.g., class designs) is described with little adequacy The use of UML diagrams and source code excerpts is poor	Problem is suitable for poster The engineering of the software is described with some adequacy. The use of UML diagrams and source code excerpts is acceptable but would benefit from revision	Problem is reasonably well-defined The engineering of the software is concisely described with much adequacy. The use of UML diagrams and source code excerpts is solid	Problem is well-defined or a good choice The engineering of the software is concisely described with considerable adequacy. The use of UML diagrams and source code excerpts is fairly good	Problem is well-defined and good choice The engineering of the software is concisely described with significant adequacy. The use of UML diagrams and source code excerpts is very effective
Presentation	10%	Delivered with no enthusiasm. The technology behind the game has not been articulated with clarity.	Delivered with little enthusiasm. Little connection with the audience. The technology behind the game has been articulated with little clarity.	Delivered with some enthusiasm, conveying technical confidence. Some connection with the audience. The technology behind the game has been articulated with some clarity.	Delivered with much enthusiasm, conveying technical confidence. Much connection with the audience. The technology behind the game has been articulated with much clarity	Delivered with considerable enthusiasm, conveying technical confidence. Considerable connection with the audience. The technology behind the game has been articulated with considerable clarity	Delivered with significant enthusiasm, conveying technical confidence and passion. Significant connection with the audience. The technology behind the game has been articulated with significant clarity.
Demo Quality	20%	There is no demo, or it is completely non-functional.	The demo demonstrates a core feature but is fairly scrappy / buggy.	The demo demonstrates a core feature but is slightly scrappy / buggy.	The demo demonstrates a core feature that works without any obvious issues.	The demo demonstrates a core feature that works without any obvious issues and aesthetically presents well.	The demo demonstrates a core feature. There is some innovation in terms of technology and/or aesthetic.