

Online Technical Portfolio

Version 2.0

BSc Computing for Games

COMP330

|  |  |
| --- | --- |
|  | Introduction Make a JavaScript portfolio on DObox.   * Something interesting as an application or game * Running from DObox * Poster * Presentation   Part A – code submission  Part B – Poster of something interesting during development  Part C – Presentation and demo of portfolio Part A Part A 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 A, upload your submission to the Learning Space. Please note, the Learning Space will only accept a single .zip file. You will receive formal feedback from your tutor three weeks after the final submission deadline. Part B Part B 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 B, upload your submission to the Learning Space. Please note, the Learning Space will only accept a single .zip file. You will receive formal feedback from your tutor three weeks after the final submission deadline. Part C Part C 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 C, attend your allotted presentation time. You will receive formal feedback from your tutor three weeks after the final submission deadline. FAQWhat 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. |
|  |  |

# Marking Descriptors: Online Technical Portfolio

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Criterion** | **Weight** | **Refer for**  **Resubmission** | **Basic**  **Competency** | **Basic**  **Proficiency** | **Novice**  **Competency** | **Novice**  **Proficiency** | **Professional**  **Competency** |
| **Threshold** | 40% | Parts A, B or C are not completed or are unsatisfactory | Submission is timely.  Attend & deliver presentation  No breaches of academic integrity | | | | |
| **Part A – code** | 20% | There are no comments, or comments are misleading.  Most variable names are unclear or inappropriate.  Code formatting hinders readability. | The code is only sporadically commented, or comments are unclear.  Few identifier names are clear or inappropriate. | The code is commented.  Some identifier names are descriptive and appropriate.  There is little obvious duplication of code or of literal values. | The code is reasonably well commented.  Most identifier names are descriptive and appropriate.  There is almost no obvious duplication of code or of literal values. | The code is well commented.  Almost all identifier names are descriptive and appropriate.  There is no obvious duplication of code or of literal values.  Some literal values can be easily “tinkered”. | The code is very well commented.  All identifier names are descriptive and appropriate.  There is no obvious duplication of code or of literal values.  Most literal values are, where appropriate, easily “tinkered” |
| **Part B- poster** | 10% | There is no poster or it does not describe the engineering of the software | The engineering of the software (e.g., class designs) is described with little adequacy | The engineering of the software (e.g., class designs) is described with some adequacy. | The engineering of the software (e.g., class designs) is concisely described with much adequacy.  The use of UML diagrams and source code excerpts is somewhat effective | The engineering of the software (e.g., class designs) is concisely described with considerable adequacy.  The use of UML diagrams and source code excerpts is quite effective | The engineering of the software (e.g., class designs) is concisely described with significant adequacy.  The use of UML diagrams and source code excerpts is very effective |
| **Part C - 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. |
| **Part D –**  **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.. |  | The demo demonstrates a core feature.  There is some innovation in terms of technology and/or presentation in the demo. |