# PRODUCTION TASKS & GAME DEMO

Version 1.2-alpha
BSc Computing for Games
COMP130

Dr Michael Scott

## Introduction

In this assignment, you will prepare a playable early-access game. Your game will be written in *C++* using *Unreal Engine 4*. You will work collaboratively in a multi-disciplinary team with other BA and BSc students.

The success of a digital game is not just determined by the fun of its design, but also by the soundness of its architecture. Put simply, games with serious bugs do not sell. As such, the value of software engineering principles cannot be emphasised enough! Reflecting upon these principles, using the lens of a development process that mirrors industry practice, is therefore critically important. Further to this, there are key quality assurance techniques and anlysis methods to appy. This will be an ongoing process, so engagement will need to persist throughout the production cycle.

This assignment is formed of several parts:

(A) **Launch**, as a **group**, a GitHub Pages website **and** promotional video, which will:
   i. **describe** the game concept;
   ii. as well as **showcase** the game's unique selling points and aesthetics.

(B) **Prepare**, as an **individual**, a draft A3 poster which will:
   i. **identify ONE** component for which **you** are individually accountable;
   ii. **describe** its requirements **and** design using UML;
   iii. and **highlight** the deliverables for the 'technical demo'.

(C) **Implement**, as a **group**, a playable game prototype which will:
   i. be a stand-alone executable suitable for play-testing;
   ii. and **illustrate** the core game mechanic.

(D) **Write**, as an **individual**, a blog post for the GitHub Pages site which will:
   i. **describe** the architecture and engineering of the game;
   ii. **analyse** the engineering quality of the game **and your** component.

(E) **Implement**, as a **group**, an early-access release canditate which will:
   i. **revise** any issues raised by your tutor and/or your peers.

(F) **Prepare**, as an **individual**, a final A3 research-style poster that:
   i. **analyses** the architectural and engineering qualities of the early-access release canditate;
   ii. and **describes** the component **you** implemented using pseudocode, code snippets, UML, **and** software quality metrics.

(G) **Present**, as a **group**, a 'technical demo' which will:
   i. **clarify** the technical content of **EACH** poster;
   ii. and **showcase** the early-access release canditate of the game.

---

*"It seems that perfection is attained not when there is nothing more to add, but when there is nothing more to remove."*

*— Antoine de Saint-Exupéry*

---

*"Good judgment comes from experience and experience comes from bad judgment!"*

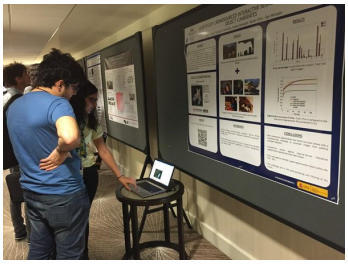
*— Fred Brooks Jr*

---

*"Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it."*

*— Brian Kernighan*

---



A poster and technical demo is a common format for presenting novel computing techniques at technology conferences such as *ACM Multimedia*.

## Assignment Setup

This assignment is a **product development task**. For this project you **MUST** use the version control repository allocated to your team.

Use the existing directory structure and, as required, extend this structure with sub-directories and branches. Ensure that you maintain the `readme.md` file.

## Part A

Part A consists of a **single formative submission**. This work is **collaborative** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) The website is live;

(b) The game is adequetely described;

(c) The unique selling points are clearly listed;

(d) The aesthetic is illustrated with at least four screenshots;

(e) The core game mechanics are showcased through a video.

To complete Part A, setup GitHub Pages and populate a single web page with information about the game and the promptional video. Show it to your product owner by the end of the first sprint.

You will receive immediate **informal feedback** from your **product owner**.

## Part B

Part B consists of a **single formative submissions**. This work is **individual** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) An appropriate game component is adequetely described;

(b) A UML use-case diagram is used to describe the requirements;

(c) A UML class diagram is used to describe the design.

To complete Part B, prepare a draft version of the poster. Submit the poster, in `.pdf` format, to the peer-review activity in the appropriate session.

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

## Part C

Part C is formed of **multiple formative submissions**. This work is **collaborative** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) A working build is available at **EACH** review;

(b) Enough work is available to conduct a meaningful review.

To complete Part C, prepare a build of the master branch ahead of **EACH** sprint review meeting. Ensure that the source code and assets have been merged. Showcase these builds at **EACH** sprint review meeting.

You will receive immediate **informal feedback** from your **product owner**.

## Part D

Part D consists of 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 the brief for details on the criteria.

To complete Part D, analyse the engineering quality of the game overall and the invididual component. Incorporate the findings into the GitHub pages site. Ensure that you include appropriate diagrams and quantitative analyses. Upload the relevant pages from the blog to the LearningSpace. Please note the LearningSpace will only accept a single `.zip` file.

You will receive **formal feedback** three weeks after the deadline.

## Part E

Part E is a **single summative submission**. This work is **collaborative** and will be assessed on a **criterion-referenced** basis. Please refer to the marking rubric at the end of the brief for details on the criteria.

To complete Part E, revise the prototype based on the feedback that you have received. Build the project and copy it to a USB storage device. Please note, you **MUST** include **both** the source code **and** a stand-alone playable build. These should be in two separate and clearly labelled folders. Instructions on how to play the game **MUST** also be included. Then, submit it physically alongside your colleagues on the other courses.

You will receive **formal feedback** three weeks after the final deadline.

## Part F

Part F 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 the brief for details on the criteria.

To complete Part F, prepare the poster using any word processing and/or presentation tool. Then, upload the relevant files to the LearningSpace. Please note, the LearningSpace will only accept a single `.zip` file.

You will receive **formal feedback** three weeks after the final deadline.

## Part G

Part G is a **single summative submission**. This work is **collaborative** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) All members of the team are present at the 'technical demo';

(b) A working **and** playable early-access release canditate is showcased;

(c) Overall, analyses of engineering quality are sufficient **and** accurate.

To complete Part G, prepare for the poster demo. Endeavour to ensure that you are comfortable discussing the content of your poster and are able to answer technical questions. Then, attend the scheduled 'technical demo' session. Please ensure that you print your poster ahead of time and bring it with you. It should be printed on A3 paper and in portrait. It is your individual responsibility to do this. Please, also ensure that one member of the team is able to setup a demo of the early-access release canditate and any equipment that you may need ahead of time.

You will receive immediate **informal feedback** from **attending staff**.

## Additional Guidance

This is a complex and challenging project. It is double-weighted in terms of module credits and so is worth a substantial proportion of your overall marks. There are **THREE** main deliverables. These are:

(i) Your blog posts, which review your key contributions and their engineering quality, submitted online as an invidual to the LearningSpace;

(ii) A working and playable version of the early-acces release canditate, submitted physically as a group on a USB device;

(iii) and, a poster highlighting **ONE** game component that you assumed responsibility for, submitted online as an invidual to the LearningSpace.

Carefully select the algorithms and/or components that you take ownership of and implement. These will not only need to interface with other game components, but also other kinds of contributions made by artistic colleagues. Aim for high cohesion and low coupling! Firmly specify the requirements and ensure they are not too broad. This will help you to avoid overlap with the work of your peers and overburdening yourself with too much work.

Please remember to commit frequently and to push your source code and related assets to the repository. This will make it easier for you to maintain a backup of your work. It will also help you to measure your productivity. Version control **MUST** be a core and essential part of your workflow, not merely a place to archive your work. You will be expected to maintain an archive of playable builds to demo your work at any time.

Poor planning and poor time management can have a substantial impact on this assignment. As some of you may have already discovered, programming is quite unlike many of the other subjects you may have studied in that it cannot be "crammed" into a last minute deluge. Sustain a steady pace across the duration of the course. Do a little programming every day, if you can!

For the most part, your work will be marked as a group effort. However, we want to avoid the situation where students try to "coast" through the assignment on their fellow group members' work, and equally the situation where one member of the group takes the lion's share of the work and prevents the others from contributing effectively. Marks will be weighted by a multiplier for **individual contribution**, which aims to penalise both of these behaviours. We assess this by several means: Does their contribution in this sprint add value to the game? Has the student effectively utilised their specialist skills? Has the student effectively utilised agile and software engineering principles? Has the student demonstrated good teamworking and communication skills? Has the student checked their contribution into souce control? Any student who has contributed their *fair share* of effort to the project will receive a fair % for their effort, so any student who is putting in the appropriate level of effort has no need to worry. Note that effort is not the same as productivity.

**Note:** In addition to sprint retrospectives and peer-review activities, your product owners will track your individual contributions using your GitHub Pages blog, the version control system, and reports generated by SparkPLUS—alongside other criteria, as described above, which they will note down in product owner meetings.

You are expected to act professionally with respect to this project. Most notably, it is **mandatory** for the team to arrange a daily stand-up meeting and it is **mandatory** for every member of the team to attend. This should be maintained every weekday throughout the duration of the project, including the studio period and days when you have sessions.

It is also **mandatory** to attend every single meeting with your product owner.

Your attendence to these meetings will be monitored. If you cannot attend a meeting, you **MUST** send a email in advance to your product owner. This should detail your apologies just as if you were emailing a senior colleague or manager in a real business context. You will be penalised on your individual contribution multiplier for any unauthorised absences.

Now that you are working in larger, multi-disciplinary teams, you will encounter collaborative complexity. There will be many more lines of communication, and therefore greater opportunity for disorganisation and conflict. Likewise, some members of the team may become agitated if others appear to be less productive than they should be. Whatever forms that these may take, it is important that you recognise such challenges and approach them in a collegiate and constructive manner. It is *critically important* that you maintain your professionalism. Be helpful and understanding. Your criticisms should be a gift to help your peers grow, improve, and mature. This means that concerns should relate to directly to working practice (or lack thereof) and be reasonably actionable. When raised, the discussion should be fair and tolerate the possibility of a compromise, as well as the your potential involvement in providing direct support for your peer.

As adults, in the first instance, you should resolve these challenges yourself. Do not permit them to fester and stagnate. Tackle these head-on at the earliest opportunity. However, if there is a breach of professional conduct, contact your product owner.

## 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/issues`.
  Please create an issue and comment accordingly.

## Additional Resources


- Stroustrup, B. (2014) Programming: Principles and Practice using C++. Second Edition. Addison Wesley.
- Keith, C. (2010) Agile Game Development with Scrum. Pearson.
- `https://www.mountaingoatsoftware.com/agile/user-stories`
- `https://literateprogramming.com`
- `http://gameprogrammingpatterns.com/`
- `https://blog.codinghorror.com/`
- `http://martinfowler.com/articles/continuousIntegration.html`
- `https://doxygen.org`
- `http://www.binpress.com/blog/2015/04/06/guide-launching-indie-games-part-three-getting-press/`
- `http://www.gamasutra.com/blogs/RogerPaffrath/20131115/204871/What_NOT_to_do_when_starting_as_an_indie_game_developer.php`

# Marking Rubric (Production Tasks — (i) Invidual Specialist Contribution & (ii) Game with Instructions)

Criteria marked with a † are weighted by individual contribution to a shared deliverable. All other criteria are individual.

| Criterion | Weight | Refer for Resubmission | Basic Competency | Basic Proficiency | Novice Competency | Novice Proficiency | Professional Competency |
|---|---|---|---|---|---|---|---|
| Threshold | 40% | Unsatisfactory participation, academic misconduct, and/or no working build delivered. | Satisfactory participation in the the collaborative project.<br>A playable build is delivered in each sprint review and final submission. | | | | |
| Value of Individual Specialist Contribution | 2.5% | Few to no individual contributions that are relevant to a computing specialism. | Contributions add some value to the project. | Contributions add much value to the project.<br>An attempt has been made to align contributions with the high concept and USPs. | Contributions add considerable value to the project.<br>Contributions are somewhat aligned with the high concept and USPs. | Contributions add considerable value to the project.<br>Contributions are adequetely aligned with the high concept and USPs. | Contributions add significant value to the project.<br>Contributions are clearly and directly aligned with the high concept and USPs.<br>Contributions have provided some support for peers. |
| Quality of Individual Specialist Contribution | 12.5% | Few to no individual contributions that are relevant to a computing specialism. | Little engineering quality. | Some engineering quality.<br>Blog posts elicit appropriate technical details.<br>UML diagrams and pseudocode are used. | Much engineering quality.<br>Blog posts elicit appropriate technical details.<br>UML diagrams and pseudocode are clear and somewhat effective. | Considerable engineering quality.<br>Blog posts leverage technical details effectively.<br>UML diagrams and pseudocode are clear and effective. | Significant engineering quality.<br>Blog posts provide clear and effective technical detail.<br>UML diagrams and pseudocode are clear and very effective. |
| Agile Working Practice | 7.5% | Working practices are unacceptable and/or agile principles have not been applied. | Little engagement with agile working practice.<br>Clear evidence of attendance at regular sprint planning meetings.<br>Clear evidence of attendance regular sprint retrospective meetings.<br>Little to no evidence of attendance at daily stand-ups. | Some engagement with agile working practice.<br>Clear evidence of attendance at regular sprint planning meetings.<br>Clear evidence of attendance regular sprint retrospective meetings.<br>Some evidence of attendance at daily stand-ups. | Much engagement with agile working practice.<br>Clear evidence of attendance at regular sprint planning meetings.<br>Clear evidence of attendance regular sprint retrospective meetings.<br>Clear evidence of attendance at daily stand-ups.<br>There is evidence of some use of CI, TDD, and bug tracking to support the project. | Considerable engagement with agile working practice.<br>Clear evidence of attendance at regular sprint planning meetings.<br>Clear evidence of attendance regular sprint retrospective meetings.<br>Clear evidence of attendance at daily stand-ups.<br>There is evidence of some use of CI, TDD, and bug tracking to support the project. | Significant engagement with agile working practice.<br>Attendence to all daily stand-ups and regular meetings.<br>CI, TDD, and bug tracking have been used effectively to support the project. |
| Collaborative Practice | 7.5% | Version control has not been used. | Material has been checked into version control at least once per sprint.<br>Communication has been sufficient. | Material has been checked into version control at least once per week.<br>Commit messages are clear, concise and relevant.<br>Communication has been somewhat appropriate.<br>Conduct and teamwork in meetings has been sufficient. | Material has been checked into version control several times per week.<br>Commit messages are clear, concise and relevant.<br>Branches are used sensibly.<br>There is a little evidence of engagement with peers (e.g. code review).<br>Communication has been effective.<br>Conduct and teamwork in meetings has been somewhat appropriate. | Material has been checked into version control frequently each week.<br>Commit messages are clear, concise and relevant.<br>Branches are used somewhat effectively.<br>There is some evidence of engagement with peers (e.g. code review).<br>Communication has been effective.<br>Conduct and teamwork in meetings has been appropriate. | Material has been checked into version control very frequently each week.<br>Commit messages are clear, concise and relevant.<br>Branches are used effectively.<br>There is much evidence of engagement with peers (e.g. code review).<br>Communication has been very effective.<br>Conduct and teamwork in meetings has been effective. |

| Criterion | Weight | Refer for Resubmission | Basic Competency | Basic Proficiency | Novice Competency | Novice Proficiency | Professional Competency |
|---|---|---|---|---|---|---|---|
| Conceptual Coherence | 10% † | No user stories and/or sprint plans are provided. | Few user stories are distinguishable and easily measured.<br>Sprint plans provide little support for the project. | Some user stories are distinguishable and easily measured.<br>Sprint plans provide some support for the project. | Most user stories are distinguishable and easily measured.<br>User stories correspond to the game design.<br>Sprint plans provide much support for the project. | Nearly all user stories are distinguishable and easily measured.<br>User stories clearly correspond to the game design.<br>Sprint plans provide considerable support for the project. | All user stories are distinguishable and easily measured.<br>User stories clearly and comprehensively correspond to the game design.<br>Sprint plans provide significant support for the project. |
| Functional Coherence | 10% † | No gameplay elements have been implemented and/or the code fails to compile or run. | Few gameplay elements have been implemented.<br>There are many obvious and serious bugs. | Some gameplay elements have been implemented.<br>There are some obvious bugs. | Many gameplay elements have been implemented.<br>There is some evidence of feature creep.<br>There are few obvious bugs. | Almost all gameplay elements have been implemented.<br>There is little evidence of feature creep.<br>There are some minor bugs. | All gameplay elements have been implemented.<br>There is no evidence of feature creep.<br>Bugs, if any, are purely cosmetic and/or superficial. |
| Creative Innovation | 5% † | There is no website or press kit, or the game does not resemble either. | Few promised gameplay elements are in-game.<br>The website and press kit have little clarity. | Some promised gameplay elements are in-game.<br>The website and press kit have some clarity. | Many promised gameplay elements are in-game.<br>The website and press kit have much clarity.<br>Promotional material evokes some excitement. | Almost all promised gameplay elements are in-game.<br>The website and press kit have considerable clarity.<br>Promotional material evokes much excitement. | All promised gameplay elements are in-gamed.<br>The website and press kit have significant clarity.<br>Promotional material evokes considerable excitement. |
| Engagement | 5% † | No insight into the appropriate use of programming constructs is evident from the source code.<br>No attempt to structure the program is evident (e.g. one monolithic source file). | Little insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is poor. | Some insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is adequate. | Much insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is appropriate. | Considerable insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is effective. There is high cohesion and low coupling. | Significant insight into the appropriate use of programming constructs is evident from the source code.<br>The program structure is very effective. There is high cohesion and low coupling. |

# Marking Rubric (Technical Demo)

Criteria marked with a ‡ are shared by the group. All other criteria are individual.

| Criterion | Weight | Refer for Resubmission | Basic Competency | Basic Proficiency | Novice Competency | Novice Proficiency | Professional Competency |
|---|---|---|---|---|---|---|---|
| Basic Competency Threshold | 40% | No individual poster and/or demo is delivered and/or no evidence of C++ having been used for the demonstrated component, or either are inappropriate. | Present at the demo. A broadly appropriate poster and tech demo are delivered in a timely fashion. Engages in conversation about the game with peers and tutors. There is no evidence of academic misconduct. | | | | |
| Poster Quality | 20% | There is no poster or it does not describe the engineering of a non-trivial game component. | The engineering of the component (e.g., class designs) is described with little adequacy. | The engineering of the component (e.g., class designs) is described with some adequacy.<br>UML diagrams and source code excerpts are present. | The engineering of the component (e.g., class designs) is concisely described with much adequacy.<br>The use of UML diagrams and source code excerpts is somewhat effective. | The engineering of the component (e.g., class designs) is concisely described with considerable adequacy.<br>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.<br>The use of UML diagrams and source code excerpts is very effective. |
| Technical Insight | 20% | No or inappropriate and/or irrelevant technical insight. | Little insight into the technical qualities of the individual algorithm.<br>Little ability to explain how the algorithm fits into the game's components and architecture. | Some insight into the technical qualities of the individual algorithm.<br>Some ability to explain how the algorithm fits into the game's components and architecture. | Much insight into the technical qualities of the individual algorithm.<br>Much ability to explain how the algorithm fits into the game's components and architecture.<br>The relevance of the contribution is justified. | Considerable insight into the technical qualities of the individual algorithm.<br>Considerable ability to explain how the algorithm fits into the game's components and architecture.<br>The relevance and value of the individual algorithm are justified. | Significant insight into the technical qualities of the individual algorithm.<br>Significant ability to explain how the algorithm fits into the game's components and architecture.<br>The relevance and value of the individual algorithm are justified.<br>The individual algorithm is somewhat important to the design of the game. |
| Demo Quality | 20% ‡ | There is no demo, or it is non-functional. | The demo demonstrates few key mechanics and interfaces. | The demo demonstrates some key mechanics and interfaces. | The demo demonstrates most core game mechanics.<br>Although there may be a backup video, at least some aspect of the demo is live using the production prototype. | The demo demonstrates all core game mechanics.<br>Although there may be a backup video, much of the demo is live using the production prototype. | The demo demonstrates all core game mechanics.<br>Although there may be a backup video, a considerable part of the demo is live using the production prototype.<br>There is some innovation in terms of technologies incorporated into the demo. |