PRODUCTION TASKS & TECHNICAL DEMO

Version 1.0
BSc Computing for Games
COMP240

Dr Michael Scott

Introduction

In this assignment, you will prepare a production prototype of your product. You will continue to work in the same manner as you have during the pre-production stage (i.e., COMP230). However, a potentially shippable product should be delivered.

The ability to scope a project and diligently execute it is highly desired in the games industry. Companies do not make money working on products; they make money by shipping products! Thus, getting a prototype to the stage where it could be submitted to a publisher or distribution platform is important. Further to this, commercial considerations do not end. Engagement with the press will need to persist throughout production, so as to ensure market interest at launch.

This assignment is formed of several parts:

(A) **Implement**, as a **group**, a prototype of the game which will:
   i. **showcase** a vertical slice of the gameplay.

(B) **Present**, as a **group**, a video demonstration of your working prototype that will:
   i. **showcase** a vertical slice of the gameplay;
   ii. **and excite** an audience of press and peers.

(C) **Implement**, as a **group**, a final production-ready prototype which will:
   i. **illustrate** the product's core features and unique selling points;
   ii. and **include** at least **one** non-trivial individual contribution from **each** member of the group.

(D) **Present**, as an **individual**, an A3 research-style poster and, as a **group**, a 'collaborative game demo' which will:
   i. **demonstrate** the final production-ready prototype of the game;
   ii. **discuss** its engineering;
   iii. **and clarify** the technical content of **each individual** poster.

"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

The *Portal* series of games started out as a student-project.

Assignment Setup

This assignment is a **product development task**.

There is no defined method of version control or repository.

If you are not working with other students, you are a group of one. As such, all references to group, collaborative, and/or teamwork activities refer to yourself as an individual. Involving BA students in such activites is optional.

Part A

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

(a) A working build is showcased to the product owner at each sprint review.

To complete Part A, showcase working builds of your prototype to your product owner and other tutors in the scheduled sprint review and crit meetings.

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

Part B

Part B 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) Press material is made available;

(b) The development team are introduced and credited for their work;

(c) Key flavour text and unique selling points are mentioned;

(d) The intended aesthetic is made clear;

(e) and the core game mechanics are illustrated with at least one video.

To complete Part B, showcase your game project at the 'Show and Tell Day' scheduled in March.

You will receive **informal feedback** from **tutors** and **peers** throughout the day.

Part C

Part C is a **single summative submission**. This work is **collaborative** and will be assessed on a **holistic** basis, according to the descriptors set out at the end of this document.

To complete Part C, revise the production prototype based on the feedback you have received. Prepare a build of the game for physical submission. Then, upload the source code 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 D

Part D 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 D, prepare an A3 technical poster on your individual technical contribution to the project. Ensure that you are comfortable with its content. Then, attend the scheduled technical demo session. Print your poster ahead of time and bring it with you. This is your responsibility. Also ensure that the team can showcase a working version of their production-ready prototype. Your tutors and your peers will play your game.

You will receive **informal feedback** from **tutors** and **peers** throughout the day.

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

Additional Guidance

Carefully select the algorithm that you will take ownership of and implement. This algorithm will need to interface with other game components, and therefore affected by work of your peers. Aim for high cohesion and low coupling! It is also important that the main requirements are firmly specified and are not too broad. This will ensure that there is little overlap with the work of your peers and help ensure that you do not overburden yourself with too much work.

Please remember to commit frequently and to push your source code and related assets to the GitHub repository. This will make it easier for you to maintain a backup of your work. It will also help you to measure your productivity. GitHub should be an essential part of your workflow, not merely a place to submit your work for feedback. 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 significant impact on this assignment. As some of you may have already discovered, programming is quite unlike other subjects 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, including but not limited to: sprint reviews; individual vivas; feedback from your peers; attribution in the source code; and GitHub commit logs. 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.

Your code will be assessed on **functional coherence**: how well the finished game corresponds to the user stories, and whether the game has any obvious bugs. Correspondence to user stories runs both ways: implementing features that were not present in the design ("feature creep") is just as bad as neglecting to implement features.

Your code will also be assessed on **sophistication**. To succeed on a project of this size and complexity, you will need to make use of appropriate algorithms, data structures, libraries, and object oriented programming concepts. Appropriateness to the task at hand is key: you will **not** receive credit for complexity where something simpler would have sufficed.

Maintainability is important in all programming projects, but doubly so when working in a team. Use **comments** liberally to improve code comprehension, and carefully choose the **names** for your files, classes, functions and variables. Use a well-established commenting convention for **high-level documentation**. The open-source tool Doxygen supports several such conventions. Also ensure that all code corresponds to a sensible and consistent **formatting style**: indentation, whitespace, placement of curly braces, etc. Hard-coded **literals** (numbers and strings) within the source should be avoided, with values instead defined as constants together in a single place. Consider allowing some literal values, where appropriate, to be "tinkered" without changing the source code, e.g. by defining them in an external file read by the game on startup.

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 Education.
- `https://www.mountaingoatsoftware.com/agile/user-stories`
- `https://literateprogramming.com`
- `http://gameprogrammingpatterns.com/`
- `https://blog.codinghorror.com/`
- `https://git-scm.com/book/en/v2`
- `http://martinfowler.com/articles/continuousIntegration.html`
- `https://travis-ci.org`
- `https://doxygen.org`
- `http://dopresskit.com/`
- `http://www.binpress.com/blog/2015/04/06/guide-launching-indie-games-part-three-getting-press/`
- `https://c9.io`
- `http://www.gamasutra.com/blogs/RogerPaffrath/20131115/204871/What_NOT_to_do_when_starting_as_an_indie_game_developer.php`

Marking Rubric (Game Source Code)

Note that this assignment is **not** marked in a criterion-based fashion. Instead, your project supervisor will assign an overall grade by considering the following descriptors in relation to your project.

Criterion	Weight	Refer for Resubmission	Novice Competency	Novice Proficiency	Professional Competency	Professional Proficiency	Expert Competency
Individual Specialist Contribution	40%	Little to no invidicual specialist conribution.	To what degree did the student achieve their goals during the course of development sprints? How consistent was the student's participation in the agile/scrum process? Does the contribution fit with the game's concept? Is the contribution adding value to the game? Has the student demonstrated their ability to work well in a team? How effectively has the student utilised their specialist skills? What level of polish has the student achieved in their contribution to the game?				
Commercia Awareness	20%	Little to no commercial viability.	Is the game's USP clearly identified? How sure is the marker that the game concept is viable? Does the work demonstrate a commercial awareness of the market the game will be competing for? Was the show and tell structured to suit the available time-frame? Did the show and tell make good use of supporting materials to help communicate the core features of the game? Does the game generate excitement?				
Game Prototype	40%	Poor software engineering, lack of validation/verification, and/or inoperable.	Is the game's scope appropriate? Is the game operating as it should? Have the tools been used to achieve appropriate technical standards of practice? Does the game address effectively its target audience? How successful is the game in creating engagement for the player? Has the game any creative distinctiveness? Is the game well presented?				

Marking Rubric (Collaborative Game Demo)

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

Criterion	Weight	Refer for Resubmission	Novice Competency	Novice Proficiency	Professional Competency	Professional Proficiency	Expert Competency
Basic Competency Threshold	40%	No poster and/or production prototype is delivered, or either are inappropriate.	A broadly appropriate poster and group tech demo are delivered in a timely fashion. Every member of the team participates and demonstrated technical knowledge about their aspect of the game. There is no evidence of academic misconduct.				
Poster Quality	20%	There is no poster or it does not describe the engineering of the software with sufficient adequacy.	The engineering of the contribution to the game is described with much adequacy (e.g., class designs). The use of UML diagrams and source code excerpts is somewhat effective.	The engineering of the contribution to the game is described with considerable adequacy (e.g., class designs). The use of UML diagrams and source code excerpts is quite effective.	The engineering of the contribution to the game is described significantly (e.g., class designs). The use of UML diagrams and source code excerpts is very effective. The aesthetics of the poster have been considered well.	The engineering of the contribution to the game is described extensively (e.g., class designs). The use of UML diagrams and source code excerpts is very effective. There is much aesthetic quality, commesurate to an distinctive demonstation.	The engineering of the software (e.g., class designs) is described extensively. The use of UML diagrams and source code excerpts is very effective. There is considerable aesthetic quality, commesurate to an impressive demo at a trade event.
Technical Insight	10%	Some to individual algorithm has been contributed, or it is trivial.	Much insight into the technical qualities of the individual algorithm. Explains how the algorithm fits into the game's components and architecture.	Considerable insight into the technical qualities of the individual algorithm. Illustrates how the algorithm fits into the game's components and architecture. Evidence that much analysis and verification has been conducted.	Significant insight into the technical qualities of the individual algorithm. Illustrates how effectively the algorithm fits into the game's components and architecture. Evidence that considerable analysis and verification has been conducted. The relevance of the individual contribution is somewhat justified.	Extensive insight into the technical qualities of the individual algorithm. Illustrates how effectively the algorithm fits into the game's components and architecture. Evidence that considerable analysis and verification has been conducted. The relevance and value of the individual contribution is justified.	Extensive insight into the technical qualities of the individual algorithm. Justifies an effective fits into the game's components and architecture. Evidence that significant analysis and verification has been conducted. The significance of the individual contribution is justified.
Demo Quality	30% ‡	There is no demo, it is non-functional, or showcases few mechanics.	The demo demonstrates the key mechanics and interfaces.	The demo demonstrates the core game loop. Although there may be a backup video, at least some aspect of the demo is live using the production prototype.	The demo illustrates the core game loop. Although there may be a backup video, much of the demo is live using the production prototype. There is a little innovation.	The demo illustrates the core game loop, making clear those intriguing aspects of the game design. Although there may be a backup video, a considerable part of the demo is live using the production prototype. There is some innovation. Playing the live demo is somewhat stable, fun, and engaging.	The demo illustrates the core game loop, making clear the uniqueness and integrity of the game design. Although there may be a backup video, the whole demo can be consumed as a live, real-time experience using the production prototype. There is much innovation. Playing the live demo is somewhat stable, fun, and engaging.