

Alcwyn Parker

Introduction

In this assignment, you will work in pairs to design and prototype a novel game controller. Your prototype should function as an input device for either one of the games being developed by students on the BA Digital Games course; or the game you developed in COMP130 last semester. Your prototype should use a hardware platform such as Arduino or Raspberry Pi etc, to convert user actions into game inputs.

Computing for games embraces the core values of the hacker movement. Experimentation, ingenuity and creativity are at the heart of everything we do. Custom game controllers are the perfect place to flex your creative flare and problem solving skills, whilst gaining invaluable experience working with hardware.

This assignment is formed of several components:

- (A) **Write**, a game controller proposal:
 - i. **describe** the game you have chosen as a basis for your interface;
 - ii. **justify** your choice of game the requirements implied by the contract;
 - iii. **outline** your concept;
 - iv. **identify** potential issues that you may encounter;
- (B) **implement**, an **iterative**, design and development process:
 - i. **push** changes to **GitHub** on a regular basis;
 - ii. **evidence** substantial progress on a **week-by-week** basis;
- (C) **Evaluate**, the work of and receive **feedback** from your **peers**:
 - i. **implement** heuristic analysis;
 - ii. **discuss** your work;
 - iii. **receive** and **reflect** on feedback from peers.
- (D) **Present**, as an **individual**, a practical demo of the game controller to your tutor that will:
 - i. **demonstrate** your academic integrity;
 - ii. as well as **demonstrate** your **individual** programming knowledge **and** communication skills.

Assignment Setup

This assignment is a **pair programming task**. Fork the GitHub repository at:

<https://github.com/Falmouth-Games-Academy/comp140-hardware>

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

Modify the `.gitignore` to the defaults for **Python**. Please, also ensure that you add editor-specific files and folders to `.gitignore`.



A twitterbot sorting a picture by its pixels.

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) research is thorough;
- (b) concept is appropriate and distinctive;
- (c) approach is considered and justified.

To complete part A, on GitHub, edit the **readme.md** file to contain a description of your proposed game controller. Your proposal should include details of any background research you have done to assess commercial viability. On Trello, create a task board that defines the key requirements (in terms of components and user stories) of the controller.

Formative submission: Arrange a meeting with your tutor to discuss your concept and task board.

Part B

Part B is a **continual formative assessment**. This work is collaborative and will be assessed on a threshold basis. The following criteria are used to determine a pass or fail:

- (a) version Control used effectively;
- (b) Sufficient progress each week;
- (c) Reflective practice.

You will build your **prototype** controller over a period of **4 weeks** utilising a fast, iterative development process. Each week should see a vast improvement in the quality of design and development working towards a shippable product to demo in the fourth week. that is, a prototype which does not have any major flaws or half-finished features that prevent it from being tested, and that can be used (even if lacking some features) as a controller in the game.

Use the forked repository to store any digital artefacts (including but not limited to design sketches, photographs, art assets, source code, electronic circuit designs).

Feedback will be given on a **week-by-week** basis.

Part C

Part C is 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) 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 C, prepare your game controller for review. It must be fully functional and integrated into your chosen game. Ensure that the source code and related assets are pushed to GitHub and a pull request is made prior to the scheduled peer-review session.

Part D

Part D is a **single summative** 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) enough work is available to hold a meaningful discussion;
- (b) Clear evidence of programming knowledge and communication skills;
- (c) No breaches of academic integrity.

To complete Part D, prepare a practical demonstration of the game controller. Ensure that the source code and related assets are pushed to GitHub and a pull request is made prior to the scheduled viva session. Then, attend the scheduled viva session.

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

Additional Guidance

Falmouth University is nationally and internationally renowned as an arts institution. Despite the fact that you are studying for a Bachelor of Science degree in a technical discipline, you are still expected to strive for the same level of innovation and creative flair as your fellow students in other departments. All assignments on this course involve a mix of technical and creative activities; this assignment is more heavily weighted towards the creative than the assignments you have completed thus far. On this assignment, a competent execution of an unimaginative idea is unlikely to achieve higher than a C grade overall, as opposed to an imperfect execution of a unique and ambitious concept ? bear this in mind when working on your design. One approach to promoting creativity is divergent thinking: generation of ideas by exploring many possible solutions. Often the most interesting ideas are subversive: they deliberately go against the accepted or most obvious solution

The history of video games is littered with failed peripherals which consumers simply did not want, which were perceived as expensive gimmicks rather than legitimate enhancements to gameplay. Your creativity should be balanced by commercial awareness: your design should be informed by your research into products that have succeeded and failed in the past, and what underexploited niches exist in the present. An A? project would be a highly divergent idea, but one that has clear commercial viability. Do not be too discouraged if you fall short of this: this is a tall order even for the professionals!

We have given you some of the materials you need: an Arduino and other useful components. You will need to add your own materials to produce a functional physical prototype. A ?Blue Peter? style prototype made from household items is fine, as is something made out of modeling clay, construction toys etc. However you should still choose your materials carefully, as overly flimsy construction may lose you marks on the functionality criterion.

You may also wish to connect electronic components such as LEDs, buzzers, photoresistors etc to the Arduino, or even use a different, more flexible hardware platform such as RaspberryPi. However you are discouraged from spending large sums of money on extra hardware, and doing so is not required to achieve a high mark. If you choose to go down this route, it is possible to purchase a RaspberryPi and other useful peripheral online for around the price of a textbook (č20 ? č30).

You should aim to demonstrate a high level of sophistication in the technical execution of your prototype. An important part of sophistication is having the insight to choose the right tool for the job: if a simpler technique fulfills all the requirements, use it. The use of unnecessarily complicated techniques, serving only to showcase one?s own cleverness, is a dangerous habit for a software developer. The sole purpose of the video demonstration is to aid moderators and external examiners, who are not present for the demo session, in assessing your work. Your video does not need to be entertaining or highly polished: a smartphone or webcam video of you or someone else using the controller is sufficient.

FAQ

- **What is the deadline for this assignment?**

Falmouth University policy states that deadlines must only be specified on LearningSpace. Please examine the assignment area where you located this document.

- **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

- Guzdial, M.J . and Ericson, B. (2015) Introduction to Computing and Programming in Python: A Multimedia Approach, 4th Edition. Pearson: New York.
- Martin, R.C. (2008) Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall: New York
- <http://guide.agilealliance.org/guide/pairing.html>
- <http://www.pairprogramming.co.uk/>
- <http://www.pythontutor.com/>

Marking Rubric

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	
Functional Coherence	5% ‡	No algorithm has been implemented successfully. The source code does not compile or there are serious logical errors.	At least one algorithm has been implemented successfully. There are many obvious logical errors, more than one of which is significant.	At least two algorithms have been implemented successfully. There are several obvious logical errors, at least one of which is significant.	At least three algorithms have been implemented successfully. There are some obvious logical errors, which are not significant. The brief has been satisfied.	At least three algorithms have been implemented successfully. There are few obvious logical errors, which are cosmetic and/or superficial. The brief has been satisfied.	At least three algorithms have been implemented successfully. There are no obvious logical errors. The brief has been satisfied.	
Sophistication	15% ‡	No insight into the appropriate use of programming constructs is evident from the source code. No attempt to structure the program (e.g. one monolithic function).	Little insight into the appropriate use of programming constructs is evident from the source code. The program structure is poor.	Some insight into the appropriate use of programming constructs is evident from the source code. The program structure is adequate.	Much insight into the appropriate use of programming constructs is evident from the source code. The program structure is appropriate.	Considerable insight into the appropriate use of programming constructs is evident from the source code. 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. The program structure is very effective. There is high cohesion and low coupling.	
Maintainability	20% ‡	There are no comments in the source code, or comments are misleading. Most variable names are unclear or inappropriate. Code formatting hinders readability.	The source code is only sporadically commented, or comments are unclear. Some identifier names are unclear or inappropriate. Code formatting is inconsistent or does not aid readability.	The source code is somewhat well commented. Some identifier names are descriptive and appropriate. An attempt has been made to adhere to the PEP-8 formatting style. There is little obvious duplication of code or of literal values.	The source code is reasonably well commented. Most identifier names are descriptive and appropriate. Most code adheres to the PEP-8 formatting style. There is almost no obvious duplication of code or of literal values.	The source code is reasonably well commented, with Python doc-strings. Almost all identifier names are descriptive and appropriate. Almost all code adheres to the PEP-8 formatting style. There is no obvious duplication of code or of literal values. Some literal values can be easily “tinkered” in the source code.	The source code is very well commented, with Python doc-strings. All identifier names are descriptive and appropriate. All source code adheres to the PEP-8 formatting style. There is no obvious duplication of code or of literal values. Most literal values are, where appropriate, easily “tinkered” outside of the source code.	
Creative Flair	10% ‡	No creativity. The work is a clone of an existing work with mere cosmetic alterations.	Little creativity. The work is derivative of existing works, with only minor alterations.	Some creativity. The work is derivative of existing works, demonstrating little divergent and/or subversive thinking.	Much creativity. The work is somewhat novel, demonstrating some divergent and/or subversive thinking.	Considerable creativity. The work is novel, demonstrating significant divergent and/or subversive thinking.	Significant creativity. The work is highly original, with strong evidence of divergent and/or subversive thinking.	
Use of Version Control	10%	GitHub has not been used.	Source code has rarely been checked into GitHub.	Source code has been checked into GitHub at least once per week. Commit messages are present. There is evidence of engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of somewhat meaningful engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of meaningful engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of effective engagement with peers (e.g. code review).	
Basic Competency Threshold	40%	At least one part is missing or is unsatisfactory.	Submission is timely. Enough work is available to hold a meaningful discussion. Clear evidence of programming knowledge and communication skills. Clear evidence of reflection on own performance and contribution. Only constructive criticism of pair-programming partner is raised. No breaches of academic integrity.					