



Assignment Brief

Academic Year 2025-26

Module code & title:	COM4023 Programming Concepts	Module leader:	Dr Rawad Hammad
Assignment No. and type:	CW1 Project	Assessment weighting:	100%
Submission time and date:	Friday 12 th December 2025 2pm	Target feedback time and date:	9 th Jan 2026

Assignment task

Your task is to form a group (three recommended) where:

- Each member of the group will select and complete one requirement specified in this brief **(40%)**. **(LO2)**
- Each member must document the development and testing of their solution by making regular commits to the team GitHub repository **(40%)**. **(LO3)**
- Each member will provide an explanation of the concepts involved in developing their solution to chosen requirement (this could be live or pre-recorded) **(20%)**. **(LO1)**

Guidelines and advice

Below is further guidance on each of the three sections of the assignment that are stated above.

Implement a solution to requirements (40%):

Choose from one of the following two codebases, which were demonstrated to you:

- Space Invaders codebase which is written in Python
- Super Mario codebase which is written in Java.

Once you have selected a codebase, you will then have to assign one member of the group to write a solution to the requirements specified below:

If your team chooses the Space Invaders codebase in Python PyGame - each team member should work towards one of the following requirements:

- Requirement: The full invaders array should include the three types of aliens. These should move from side to side, and fire at the player at random. Their speed of movement should increase as the invaders are reduced in number (from being shot).
- Requirement: The defender (player) should be able to shoot multiple bullets continuously. Implement collision detection so that invaders are destroyed when successfully shot by the player, and that the player loses a life and respawns (until run out of lives) when shot by an invader.
- Requirement: The four green barriers should diminish every time they are shot by both the invaders and the defender/player.

If you choose the Super Mario codebase in Java Greenfoot, then each team member should work towards one of the following requirements:

- Requirement: Implement the tilemap of a single level as an array of objects. You should be able to load in the complete tilemap at the start of the game and then move the camera view to show upcoming parts of the level as Mario moves left and right.
- Requirement: Implement Mario's movement (left, right, and jump physics), as well as collision detection between Mario and all other tile objects – the ground tiles, blocks he has to jump over, powerups he collects etc.
- Requirement: Implement a final boss type character (e.g. Bowser) which moves in selective patterns, fires fireballs at Mario, and has animation and sound effects.

Documentation and Testing (40%)

For your chosen requirement of your chosen codebase, you are required to document the progress you make towards the solution. Requirements are designed to have individual components so they can be worked on and tested independently. However, to present a fully finished working game at the end, you will also need to connect your code to the code written for the other requirements. So, allow time near the end of the process for harmonising your solutions to work together.

You should make regular commits to your team GitHub repository, to show progress on writing the code, as well as testing this. Your team should make use of the 'wiki' and 'issues' features to document your progress. For submission purposes you will be required to document which team member was assigned to each requirement. For this component you will be scored as a team (so you all receive the same mark for this component, unless there is a compelling reason to award different marks here). For the development and demonstration components, you will be marked individually on your work.

Explanation / Demonstration of your solution (20%):

You will be required to attend a timeslot during Week 10 of the module to give a short explanation of how you solved the requirement you selected, as well as an explanation of the programming concepts involved in your solution. You should come prepared to show your solution (so have the code and documentation ready on a laptop). Each team member should aim to explain and demonstrate your solution within five minutes.

If you are registered with the Disability and Inclusion Services or have a disability, please contact your module leader to discuss how your reasonable adjustments will be applied to this assignment.

Getting Support for Your Assignment

Your first point of advice should be the person setting the assignment or the module leader.

If you need more general support with your studies (e.g. academic writing, referencing, critical thinking) then the [Student Learning and Achievement Team](#) can support development of your academic skills.

This assignment has been designed to provide you with an opportunity to demonstrate your achievement of the following module learning outcomes:

LO1: Define fundamental programming concepts.

LO2: Utilise problem-solving skills to develop a working solution to a brief.

LO3: Collaborate on and manage a team project with version control.

Practicalities: Referencing, presenting and submitting your work

Please see your programme handbook for information on the required referencing style, any presentational requirements and the process for submitting your work.

- **Explanation and demonstration** of your solutions to the requirements stated above.
- **Each member of the team should submit a document to Turnitin that contains:**
 - **Source code to your solution** (for similarity check).
 - Either the Team's **GitHub URL** or your own **GitHub URL** containing:
 - **Commit history** for developing the solution.
 - A **Wiki** which documents the development and testing of the solution.
 - Keep the **Issues** raised in your GitHub repository.

Confidentiality

You should maintain and respect confidentiality in relation to the protection of personal, technical and/or commercial information of a sensitive nature in your assessed work.

For further information and guidance, please see the University [Academic Confidentiality guidance](#).

Academic integrity

Academic integrity means taking responsibility for your own work and not acting in a way to give you an unfair advantage over another learner. This includes writing in your own words and acknowledging sources you have used. It also covers the appropriate use of Artificial Intelligence.

When you submit an assignment, you are making a declaration that it is your own work and that you have acknowledged the contribution of others and their ideas in its development (for example, by referencing them appropriately).

For further information, please see the following student facing guidance on the University website:

[BNU Student Academic Integrity Guidance](#)

[BNU Artificial Intelligence Guidance for Students](#)

There are also links to resources to support your understanding of Academic Integrity within your Blackboard module sites in the Assessment Information area.

Categorical Mark and Grade	0,1,10,20, (F)	32,35,38% (E)	42,45,48% (D)	52,55,58% (C)	62,65,68% (B)	72, 75,78% (A)	82, 85, 88, 92, 95, 98, 100% (A+)
Pass status	Fail Learning Outcomes have not been met	Marginal Fail Learning Outcomes have not been met	Pass - Satisfactory Learning Outcomes have been met	Pass – Good Learning Outcomes have been exceeded	Pass – Very Good Learning Outcomes have been exceeded	Pass – Excellent Learning Outcomes have been exceeded	Pass – Outstanding Learning Outcomes have been exceeded
Coding / Version control (40%) (LO2) This is evaluated by: <ul style="list-style-type: none"> • GitHub • Source code • Code quality 	The solution (if one submitted at all) does not fulfil the requirements specified in this brief. There is no evidence that students have investigated why the program does not compile/run.	There is evidence that an attempt has been made, but the source code does not fulfil the basic requirements of this brief. There is minimum to no evidence that the solution has been tested.	The solution does compile and create a functioning program that fulfils the basic requirements, and there is evidence that these basic features have been tested.	The solution compiles and fulfils at least half of the stated requirements (more than the basic requirements). These features implemented have supporting test evidence.	The solution compiles and fulfils most (not all) of the stated requirements. The implementation of these features is supported by appropriate test evidence.	The solution compiles, works without exception and fulfils all the requirements and evidence is presented of these features being fully tested.	In addition to a complete and functioning solution, a feature (or several) has been added that makes the game 'stand out.' There is evidence that all functionality (including the new features) has been fully tested.
Documentation (40%) (LO1, LO2) This is evaluated by: <ul style="list-style-type: none"> • GitHub wiki • GitHub issues • Design • Testing 	An empty wiki or blank template has been supplied (but hasn't been modified). Minimal commits also present.	No wiki is present with the repository. Or if a wiki has been provided, then minimal commits (1 or 2) have been made.	A wiki is present and features some detail about the development of the application. Significant sections are missing or incomplete. Few commits have been evidenced.	The wiki documents roughly half the development of this application. A reasonable number of commits have been made to the repository.	The wiki documents most of the development journey of the application but has one or two omissions. The work is strongly documented by evidence.	The development of the application has been detailed in the wiki, but there are one or two minor issues with what has been presented.	The development of the application has been extensively detailed featuring comprehensive commentary and discussion of features. No issues were found, and outstanding features have been fully documented.
Presentation/ Demonstration (LO3) (20%) This is evaluated by: <ul style="list-style-type: none"> • Presentation • Live or pre-recorded demonstration 	The student has not attended the presentation or has made an insufficient contribution to the presentation.	The student has made a small contribution to the presentation, but it was not deemed to be sufficient to pass. This is either due to poor delivery (lack of coherence and pace too fast, reading from slides and not engaging the audience, technical errors).	The student has contributed to the presentation. Their part was acceptable, but the delivery could be improved by working on several areas outlined in the criteria.	A good presentation which fulfils several criteria, however, improvements to several other criteria have been identified.	A sound presentation which fulfils many of the criteria set out. However, one or two improvements have been identified which could have made the presentation stronger and more impactful.	An excellent presentation which gives the impression of a well-rehearsed and crafted explanation of their contribution to the project. There are only one or two minor points in style, delivery or the technical understanding that could be critiqued.	This presentation meets and exceeds the required standard. It is a truly remarkable and captivating presentation that stands out.