

FIT3077 Software Engineering: Architecture and Design

Sprint One (20%) Specifications Due: 12pm Monday, 3 April 2023

Faculty of Information Technology, Monash University
Semester 1, 2023

General Project Technical Requirements and Expectations

There are several expectations, conditions and restrictions that apply to all project teams throughout the entirety of the project (i.e. all sprints).

1. The *object-oriented* programming languages you are approved to use for the project (throughout all of the sprints) are Java, Python, C++, C#, TypeScript, Smalltalk and Eiffel.
2. The application you will eventually develop **must be implemented as a standalone application** that is able to **run locally** on a single device and does not require separate server-side code to be written.
3. All work done (from Sprint 1 onwards) must be committed and pushed to the official Monash GitLab repository provided to each project team at the following URL: <https://git.infotech.monash.edu/fit3077-s1-2023/<YOUR-MOODLE-GROUP-NAME>/project>
4. Unless explicitly stated otherwise (in writing) by the teaching team, any work done, committed and pushed to any personal Git repository or only available on personal working environments will **not** be accepted for assessment. Work should be done consistently throughout each sprint, and not started near the due date and rushed through.
5. Each member of the team must contribute to the design and implementation of all deliverables, and **must not work in silos**. **Teams must not delegate entire portions of a particular task or type of deliverable (e.g., user stories) to a single team member**. For example, in Sprint 1, each member of the team is expected to contribute to, and produce some user stories, some architecture design, and some UI design, and a single member must not bear the sole responsibility of creating all user stories.
6. Although each project team is offered the flexibility in choosing technologies that they will use for the project, all teams need to be able to demonstrate how they would be able to adequately and properly apply object-oriented design and architecture with their chosen technologies. The teaching team reserves the right to make the final decision whether or not a team's choice of technologies for the project is suitable/acceptable.
7. Any issues that arise which will potentially affect the progress or performance of your team **must** be raised as soon as possible to the teaching team.

Project Overview, Description and Requirements

Monash University is planning to run an exhibition during Open Day in August that showcases student talent from various faculties, including the Faculty of IT. All prospective students, their family members and other visitors are all welcome to attend. Additionally, the Faculty of IT has invited a guest of honour by the name of Dr. Mills, an influential expert in object-oriented design, to the exhibition. Rumour has it that Dr. Mills is an avid fan of traditional games and hence, the faculty wants to ensure that there is something showcased in the exhibition that will be of interest to them.

After an extensive planning and negotiation process, the faculty has decided to approach your team to help **develop a 9 Men's Morris (9MM) game client application**. For this project, your team has also been given **some flexibility in terms of how the final game client will look like and what technologies will be used**, but the game needs to be completely developed and ready by mid-June. The faculty would like to check-in with your team regarding the progress of the game client development every few weeks so that everything remains on track and any issues hindering the timely completion of this project can be mitigated as quickly as they arise.

Since Dr. Mills is particularly passionate about object-oriented design and there is a possibility that Dr. Mills might want to speak to your team regarding how the game client has been developed, the faculty has also requested that the game client be developed with proper software development practices and object-oriented approaches/principles in mind throughout this project.

The faculty would like a standard implementation of the 9MM game and would like your team to implement additional functionality, as follows:

1. **Basic Requirements for Basic Prototype (Sprint 3 deliverable):** The game client must be fully **able to play 9MM according to the standard rules**, and the game should be **playable within the same game client instance by two players that will take turns** in making their respective moves.
2. **Advanced Requirements for Final Prototype (Sprint 4 deliverable):** Select ONE of the following advanced requirements to add to the Basic Prototype. **Teams of 4 members will need to implement any TWO** of the advanced requirements.
 - a. **Considering that visitors to the student talent exhibition may not necessarily be familiar with 9MM, a tutorial mode needs to be added to the game. Additionally, when playing a match, there should be an option for each player to toggle "hints" that show all of the legal moves the player may make as their next move.**
 - b. Players are allowed to undo their last move and the game client should support the undoing of moves until there are no more previous moves available. The game client also needs to be able to support saving the state of the currently active game, and be able to fully reload any previously saved game(s). The game state must be stored as a simple text file where each line in the text file represents the current state of the board and stores information about the previously made move. It is anticipated that different file formats will be required in the future so any design decisions should explicitly factor this in.

- c. A single player may play against the computer, where the computer will randomly play a move among all of the currently valid moves for the computer, or any other set of heuristics of your choice.
- d. Allow a game of 9MM to be playable using two different game client instances - one made by your team and the other made by another team. Moves made on one game client need to be able to appear on the other game client as soon as they are made.
- e. Develop a 3D version of the 9MM game. This advanced functionality is only recommended for 4 people teams. Teams wishing to attempt this should discuss with the teaching team by the end of Sprint 2.
- f. If you have a brilliant and equally 'sized' functionality idea for the additional advanced feature, you can pitch it to the teaching team. Additional advanced functionality not listed above must be discussed with the teaching team and approved in advance, e.g. by end of Sprint 2.

Sprint 1 Deliverables (20% of final unit mark)

All tasks are mandatory.

1. Team Information

Document the following pieces of information related to your team.

- *Team Name and Team Photo*
 - Come up with *your own personal team name*. Your team name must be professional. **The name of the team you belong to in Moodle (with the format *<campus>_<workshop session>_<team number>*) is not an acceptable team name for this task.**
 - Your team photo must not be edited/photoshopped. All team members in an on-campus group must be present together physically at the time of taking the photo. For online groups, a Zoom team photo is accepted. **Team photos via Zoom will not be accepted for on-campus groups.**
- *Team Membership*
 - Document the basic information of *each team member* - for example *name and contact details*.
 - List out what the *technical and professional strengths of each member* are.
 - Provide a *fun fact about each member* that not many people know about.
- *Team Schedule*
 - Document your team's *regular meeting schedule* and *regular work schedule*.
 - Document *how the workload will be distributed* and *managed* within your team.
- *Technology Stack and Justification*
 - Document what *programming languages, APIs, and technologies* are you planning to use and *how this maps to the team's current expertise*, and *which ones you anticipate needing support from your tutors with*.
 - *Justify your team's final choice* of technologies that will be used.

2. User Stories

Submit a list of user stories (e.g., 10 to 25 stories) that covers both the basic 9MM gameplay and the chosen advanced requirements specified above. A majority of the user stories are expected to be devoted to the basic requirements for the Basic prototype.

If your group consists of 4 members, your user stories must also cover the additional advanced requirement.

3. Basic Architecture

Design and draw a domain model that covers both the basic 9MM gameplay and the chosen advanced requirements specified above.

Provide detailed justifications for the domain model that you come up with, with a focus on the following aspects:

- Rationale for each chosen domain and their relationships (if any)
- Were there any design choices that you had to make while modelling the domain and WHY?
- Explain any assumptions you have made, as well as any other part of your domain model that you feel warrants a justification as to WHY you have modelled it that way.

If your group consists of 4 members, your domain model and justifications must also cover the additional advanced requirement.

4. Basic UI Design

Draw low-fidelity (low-fi) prototype drawings of the proposed user interface for the application. The low-fi prototypes need to demonstrate both the basic 9MM gameplay and the chosen advanced requirements specified above. The prototypes should cover all the key interaction scenarios, e.g. initial board, placing tokens, moving tokens, 'flying', forming a mill (win condition), and the advanced feature of your choice. This can be achieved in one large drawing space or across multiple pages. Avoid redundancy, i.e. do not create multiple prototypes for the same interaction. All drawings should be large and clear enough to understand and any writing should be legible. You may use pen and paper, or digital drawing tools.

If your group consists of 4 members, your lo-fi prototype drawings must also cover the additional advanced requirement.

Recording Sprint Contributions (hurdle)

Each team is required to have a single contribution log page for all team members in the **wiki section** of the team's project in Monash GitLab that documents the contributions of each member towards the particular sprint. The link to the wiki section of each team is as follows:

<https://git.infotech.monash.edu/fit3077-s1-2023/<YOUR-MOODLE-GROUP-NAME>/project/-/wikis/>

Each team member should update the contribution log by adding entries that describe the pieces of work said team member had done, the **start time** and **date**, **amount of time spent**, **who performed the work** and if applicable, any **other notes**. Each team member is not expected to immediately record each piece of work into the contribution log as soon as it is done, but is expected to record into the contribution log all of the unlogged work done thus far **at least once or twice per week**. **You SHOULD NOT update the contribution log on behalf of other team members.**

Notes

- *The entirety of the sprint submission will be marked holistically and all deliverables are considered during assessment of submitted work.*
- For the submission to be considered complete, **rationales must be submitted** as part of all sprint deliverables that require them.
- The **rationales submitted in Sprint One will not carry any marks**. Instead, formative feedback on the rationales submitted will be provided. This will be followed by a workshop on how to write good rationales during one of the upcoming workshops. ***Subsequent rationales submitted in Sprints Two to Four will be marked.***

Submission Instructions

Monash GitLab Repository

- As mentioned in the *General Project Technical Requirements and Expectations* section, **all project work** needs to be pushed to your team's provided Monash GitLab repository.
- If you use any software tools to create any of your wireframes, diagrams or word documents, you **MUST** export the wireframes and diagrams **as JPEG, PNG or PDF file(s)** and the **word documents as PDF file(s)**. Ensure these exported files are then pushed to your team's repository **in addition to the raw files** from these tools. Otherwise you may lose marks for not providing your deliverables in a correct and/or readable format.
- Neat and legible hand-drawn diagrams are acceptable but must be scanned at a high resolution and also pushed to your team's repository.
- For the purposes of final submission, all text-based/written/drawn deliverables **can be left as individual JPEG/PNG/PDF documents**, or optionally compiled into **a single PDF** document and subsequently pushed to the repository.

Moodle

- In addition to all project work being pushed to Monash GitLab by the due date, your project work **must also be submitted by the due date to Moodle** by **downloading** the current state of your **repository** from Monash GitLab **as a zip** file and then submitting it.
- **Only one team member** (on behalf of their project team) needs to make the submission to Moodle. Submissions must be in the **fully submitted state - drafts will NOT be accepted**.
- When submitting, please ensure that all files you intend to submit are included in the submission and your submitted file(s) can be properly opened/extracted/read before clicking submit.

Sprint Contribution Log

- The contribution log must be a *wiki page* in the team's Monash GitLab project repository: <https://git.infotech.monash.edu/fit3077-s1-2023/<YOUR-MOODLE-GROUP-NAME>/project/-/wikis/>
- **Each team member must** record details of their **own contributions** towards the sprint in their team's contribution log. This must be done **at least once or twice a week** and a team member **should not** update the log on behalf of other team members.

Tools

The following tools can be used in support of making the sprint deliverables. Diagrams/wireframes created with other tools such as (but not limited to) *Lucidchart*, *diagrams.net*, *Figma*, etc. are acceptable. For diagrams involving UML notation, the UML 2.5.1 standard needs to be followed.

- Lucidchart: <https://www.lucidchart.com>
- Diagrams.net: <https://www.diagrams.net/>
- Figma: <https://www.figma.com>
- UMLet Standalone: <https://www.umlet.com>
- UMLet Online: <http://www.umlet.com/umletino/umletino.html>
- UMLet VS Code Extension:
<https://marketplace.visualstudio.com/items?itemName=TheUMLetTeam.umlet>
- UMLet Eclipse Plugin:
<https://marketplace.eclipse.org/content/umlet-uml-tool-fast-uml-diagrams>
- Visual Paradigm Online: <https://online.visual-paradigm.com>

Further Notes

Extensions

No extensions will be given in normal circumstances. An extension may be granted in special circumstances as per faculty policy. Extensions must be applied online at the following link: <https://www.monash.edu/exams/changes/special-consideration>. For any queries related to the assessments, please contact:

- if you are at Clayton campus, the Clayton team at <fit3077.clayton-x@monash.edu>
- If you are at Malaysia campus, Chong at <chong.chunyong@monash.edu>

Lateness

Penalty of **10% of the total available marks per day late or part thereof** after the due date, **including the weekends**.

Authorship

The work in this assessment is team-based and the final submission must be identifiable as a team's own work. Breaches of this requirement will result in submitted work not being accepted for assessment and may result in disciplinary action. Refer to the *Academic Integrity and Plagiarism/Collusion Section* that follows for more details.

Academic Integrity and Plagiarism/Collusion

Academic integrity is about the honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

To fully acknowledge artificial intelligence models (such as Generative AI) which cannot be cited, you should include both a declaration of the generated material and a declaration of the technologies that were used. At a minimum, you should include a declaration of use that explains what technologies, if any, you have used to generate material in working on your assessment.

For more information about acknowledging the use of generative AI for assignments, please refer to the following link:

<https://www.monash.edu/learnhq/build-digital-capabilities/create-online/acknowledging-the-use-of-generative-artificial-intelligence>

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own. Monash University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source;
- Copyright material from the internet or databases;
- Collusion between students.

For further information on our policies and procedures, please refer to the following link:

<https://www.monash.edu/students/study-support/academic-integrity>

Assessment Guide

	Band				
	Poor	Acceptable	Good	Very Good	Excellent
Team Information	A team name given but no photo provided; team members listed but not introduced; no meeting scheduled listed; no technology to be used provided	A team name given but no photo provided; team members listed but not introduced; no meeting scheduled listed; very basic information about technologies used but not justified.	Team name and team photo included; list of team membership given, but only basic introduction of team members; basic meeting schedule provided; technology stack listed but only some justification provided	Professional team name and team photo included; list of team membership, but only basic introduction of team members; basic meeting schedule provided; technology stack listed and reasonable justification provided (but no alternatives discussed)	Professional team name and team photo included; complete list of team membership, all team members properly introduced, and meeting schedule provided; technology stack discussed, justifications provided, discarded alternatives discussed
User Stories	User stories written to cover some Game functionality; without consideration of the recommended format or the INVEST acronym; with redundancy.	User stories written well to cover the key Game functionality; applying some aspects of the INVEST acronym.	User stories written in the recommended format; covering the complete Basic game functionality; applying most aspects of the INVEST acronym; without redundancy.	User stories written in the recommended format; covering some 'user' roles; and the complete Basic game functionality; applying most aspects of the INVEST acronym; without redundancy.	User stories written in the recommended format; covering multiple 'user' roles; and the complete Basic game functionality; applying all aspects of the INVEST acronym; without redundancy.

Architecture	Incomplete Domain Model (missing entities); using wrong notation; no explanations or rationale provided	List of domain entities provided, but domain model lacks key entities and/or associations; correct notation used; very basic explanations and/or rationale for model provided	Domain Model mostly complete with only few associations and/or cardinalities missing; correct notation used; explanation for some domain entities given; some rationale for chosen model given (but yet incomplete)	Complete Domain Model; correct notation, including most cardinalities; explanation for some domain entities given; some rationale for chosen model given (but yet incomplete)	Complete Domain Model; correct notation, including cardinalities for all associations; explanation for each domain entity and each association given; rationale for chosen domain model provided and discarded alternatives listed
UI Design/ Low Fi Prototype	A low-fi prototype; covering some elements of the 9MM game; unclear or unreadable images; not capturing the game elements; with redundancy.	A clearly drawn low-fi prototype; covering some elements of the 9MM game.	A clearly drawn low-fi prototype; covering most key elements of the 9MM game; without redundancy.	A clearly drawn low-fi prototype; covering all elements of the 9MM game; without redundancy.	A clearly drawn low-fi prototype; covering all elements of the 9MM game; without redundancy; with creativity in token or board design;