

Teaching Period 3a, 2024

IFQ563 Object Oriented Design

Assignment 2: Final design report and implementation

Word limit:

- Project design report: 5–10 A4 pages.
- Implementation source code: ZIP file with ~500 lines of code compatible with Microsoft Visual Studio 2022.

Weighting: 70%

Due date: 11:59pm AEST Friday 16 August 2024 (Assignment Week)

After you have read this information, head over to the [Assignment Q&A](https://canvas.qutonline.edu.au/courses/1658/discussion_topics/13933) (https://canvas.qutonline.edu.au/courses/1658/discussion_topics/13933) discussion board to ask any questions and see what your peers are saying about this assignment.

Assignment overview

In this assignment you will create and document an object-oriented design and implement it using C# for a larger real-world motivated problem.

You will use feedback from your [Assignment 1: Preliminary design](https://canvas.qutonline.edu.au/courses/1658/assignments/7795) (<https://canvas.qutonline.edu.au/courses/1658/assignments/7795>) to refine and implement your object-oriented design. However, you are advised to begin implementing your design even before receiving feedback on your preliminary design.

The objective of this assignment is to gain practical experience with object-oriented software development through object modelling with design diagrams and applying standard software design principles and patterns. This understanding will be demonstrated by creating and documenting an object-oriented design and implementing it using C# for a larger real-world motivated problem.

This assignment is a continuation of Assignment 1: Preliminary design (<https://canvas.qutonline.edu.au/courses/1658/assignments/7795>)

. Please ensure you familiarise yourself with the IFQ563 Assignment specifications (DOCX 130 KB) (<https://canvas.qutonline.edu.au/courses/1658/files/444786?wrap=1>)

(https://canvas.qutonline.edu.au/courses/1658/files/444786/download?download_frd=1) before continuing. This assignment is individual work, and all final tasks are to be submitted together through Canvas. While you will discuss the assignment with your OLA and peers, your submissions must be your original work.

This assignment supports unit learning outcomes 1, 3, and 4.

The use of artificial intelligence tools (for example Chat GPT) is not permitted in any assessment in this unit. The use of such tools when not authorised may be treated as a breach of MOPP C/5.3 Academic integrity (https://www.mopp.qut.edu.au/C/C_05_03.jsp) and appropriate penalties imposed.

Assignment details

In this assignment, you will create and submit a project design report in PDF format and your implementation source code in a ZIP file containing all of your C# project files.

Project design report

The purpose of the project design report is to document an explanation of your design, including a statement of completion, an overview of your final design, detailed design documents, identification and a brief justification of design principles and patterns, instructions on how your program can be executed, and a summary of classes/interfaces to be reused from existing libraries.

Your report will be marked for the simplicity and elegance of design. It is in your best interest to make your design as neat, clear and understandable as possible. In general, your detailed design should speak for itself, and you should only include explanations of your design where your intentions may not be clear. Still, the use and justification of design principles and patterns should mostly cover this. Use the following list to structure your design report. Page limits indicated are a guide only, however, your total submission should not exceed ten pages.

- **Statement of completion:** Clearly declare the requirements that have and haven't been completed (half page).
 - **Overview of your final design:** Provide a summary of changes from the preliminary design, including what and why changes were made (one page).
 - **Detailed design documents:** Include a class diagram, an object diagram and a sequence diagram (maximum of three pages). You don't need to provide CRC cards because they are reflected in your final class diagram.
 - **Design principles and patterns:** Identify and briefly justify the design principles and patterns used. For each design pattern, you should indicate the participating classes (in design diagrams) and justify their use in your design with a few sentences (one to two pages).
 - **Execution instructions:** A brief document on how your program can be executed (at most one page).
 - **Summary of classes/interfaces:** A short summary of classes/interfaces to be reused from existing libraries and frameworks. For example, if you use the Collections library, list the classes you will use without further explanation (one page).
- You must submit** a project design report in PDF format with:
- no more than 10 A4 pages in length
 - 2 cm margins on all sides
 - 12-point Times New Roman font
 - single spacing.

Implementation source code

- You must submit a working implementation including full C# source code. You should zip all your project files and upload this ZIP file to Canvas when you submit your design report.
- You do not need to provide a user guide or tutorial, but you must clearly document how your program can be executed in your design report.
- Your class implementations must be strictly faithful to the documented class designs in your final design report. That means the classes in the source code must correspond to the same classes defined in the class diagram, including their properties, operations and relations to other classes.
- The submitted project files will be compiled and executed with .NET 6. You must ensure your submitted code can be compiled and run properly with the correct .NET version.
See [0.4 Technical requirements](https://canvas.qutonline.edu.au/courses/1658/pages/0-dot-4-technical-requirements?wrap=1) (<https://canvas.qutonline.edu.au/courses/1658/pages/0-dot-4-technical-requirements?wrap=1>) for a free link to Visual Studio 2022 (VS2022), which supports .NET 6.
- Unfaithful class implementations will receive zero for implementation. Uncompilable or inexecutable source code cannot be marked and will receive zero for implementation. To confirm the version of .NET on the computer, simply run the following command:
dotnet --version
- To check that your project code can be compiled and executed on .NET 6, open the folder containing the project file (.csproj) and run the following commands:
dotnet build
dotnet run



Supporting resources

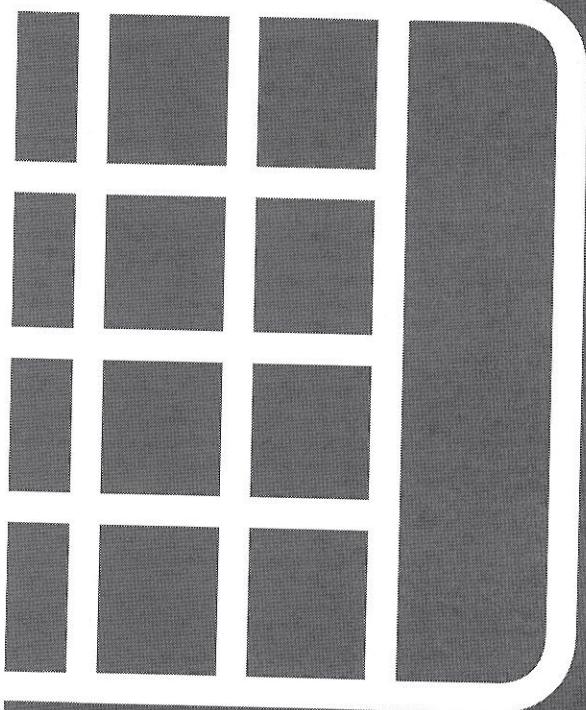
- [QUT cite|write: APA](https://www.citewrite.qut.edu.au/cite/qutcite.jsp#apa) (<https://www.citewrite.qut.edu.au/cite/qutcite.jsp#apa>).

Assignment criteria

- 1. Design report: Design quality.
- 2. Design report: Use of design patterns.
- 3. Design report: Documentation.
- 4. Implementation: Fulfilment of requirements.
- 5. Implementation: Quality of code.
- 6. Implementation: Usability of application.



RUBRIC



Rubric

Select the 'Rubric' icon for more details about this assignment's marking criteria.



- ▶ [Assessment declaration](#)
- ▶ [Submission details overview](#)
- ▶ [Request Assignment Extension](#)

Search entries or author...

All



Your OLA will be checking in on this discussion, and providing feedback and support.

Long Ngo

Update #3:

Hi everyone,

As we have just finished Assignment 1, I wanted to remind you all of the importance of **designing a framework for 2-player board games** in your assignments. Specifically, this means creating a set of classes that provide a **core structure for any 2-player board game, without being tied to any specific game**.

For the TicTacToe game, you should extend this framework by **adding several additional classes that provide functionalities specific to TicTacToe**. The core structure and common functionalities should already be provided by your framework.

If you decide to implement another game, you would follow the same process: add classes specific to the new game while reusing the framework. This demonstrates the ideas of extensibility and reusability. **Design patterns are the tools that helps us to achieve those goals.**

Please let me know if you have any questions.

Cheers,



Long Ngo

Hi everyone,

Update #2: As we are finalizing Assignment 1 and moving on to Assignment 2, which involves implementing a game:

1. It is normal for the design in Assignment 2 to be different from the one in Assignment 1. The idea behind Assignment 1 is to gather feedback and make design adjustments.
2. Since Assignment 2 includes a game implementation, please ensure that your report highlights the inter-class relationships between the classes of the game and the classes of the framework. This will help in understanding how to extend and reuse the framework.

Cheers,

Long



Long Ngo

Hi everyone.

Update #1:

The following screenshots are examples of how a game would interact. Note that this is just an example, it could be improved and you do not have to follow it.



```
Choose a name for the first player [default: Player 1]: A
Choose a name for the second player, leave empty for a computer player [default: ]:
Welcome to Wild Tic-Tac-Toe, each player takes a turn placing a X or O in a cell in
attempts to be the last to connect 3 in a row and win the game.
```

	1	2	3
1			
2			
3			

```
A (type HELP for a list of commands)> |
```

A (type HELP for a list of commands)> HELP
Commands:
MOVE - Play a turn
UNDO - Undo the last turn.
REDO - Redo the last undone turn.
SAVE - Saves the game state to file.
LOAD - Loads the game state from file.
EXIT - Exits the game.

A (type HELP for a list of commands)> MOVE
A, your turn. Enter the row (1 to 3) followed by the column (1 to 3)
and then the symbol (X or O) [example: 1 1 X]: 1 1 O

	1	2	3
1	0		
2			
3			

Computer (type HELP for a list of commands)> |



A (type HELP for a list of commands)> MOVE
A, your turn. Enter the row (1 to 3) followed by the column (1 to 3)
and then the symbol (X or O) [example: 1 1 X]: 1 1 X
Invalid move. Cell already occupied.
A, your turn. Enter the row (1 to 3) followed by the column (1 to 3)
and then the symbol (X or O) [example: 1 1 X]: 1 2 X

	1	2	3
1	0	X	
2			
3		X	



A (type HELP for a list of commands)> MOVE
A, your turn. Enter the row (1 to 3) followed by the column (1 to 3)
and then the symbol (X or O) [example: 1 1 X]: 2 2 X

	1	2	3
1	O	X	
2		X	
3		X	X

The game is over and player A is the WINNER, well done!

