# CCT College Dublin

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | *Concurrent Systems* |
| **Assessment Title:** | CA2 |
| **Lecturer Name:** | *Sam Weiss* |
| **Student Full Name:** | Guilherme Felix Carlini de Lima |
| **Student Number:** | 2021309 |
| **Assessment Due Date:** | 02/06/2024 |
| **Date of Submission:** | 02/06/2024 |

## Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

# AI Prompts

Here I’ll provide some of the prompts that was crucial to create and develop the game.

“Can you help me break down this requirements so I could plan how to create this game”

After reading what the AI gave me, My plan to start developing the game was, start with the Board, having a display of the board then I added some monsters into a 2D array, so I could test it. Then create a function to move the monsters into the array, having that, create a way, via console, to add monsters into the array, but following the requirements and with that having a way to visualize the game so I could start planning what would be the next step.

* “Here is my code in console app, how do I move the monster from the row x to row y”
* “how do I create a function to add monsters into the grid”
* “for this movement function please apply these movement rules: monster can move 10 squares in horizontal and vertical, but only 2 in diagonal.”
* “Help me develop a function where it checks if this monster wins against other, and also check if this monster is mine or not”
* “Help me develop a function where it checks if there is no enemy on the way to my desired location”

Basically most of my prompts at the beginning of the development was to make the game, based on the requirements. Once I had a functional game in my console I could move to the web project.

I had so many issues with the C# language in general, so I decided to move to JavaScript. To don’t lose my progress or everything I’ve done there, I simply asked to “convert” from C# to JavaScript. So I could continue with my plan in JavaScript.

* “Based on the code provided, could you transform this code into Node.js code, with a HTML file to display the board of the game.”
* “create the CSS grid based on this 2D Matrix”
* “I would like to create a game where I click into the monster and move to some other part of the grid”
* “Can you help improve my code to create a turn system?”
* “Is this logic from my player stats is correct?”
* “can you give me an example of how to create a function to display the current player of the match?”
* “How to create a function of round system?”
* “Can you transform my Index.html into dark theme”

# Self Reflection

This project I had so many ups and downs along the way. I started this project trying to create this whole system using Spring Web, in Java, however I didn’t have much success, so one day I was chatting with some friends and they recommended me to do it in C#, ASP.net, Blazor and as far I heard that C# language is quite similar to the Java, I felt that would be challenging to get to learn about something new. They guided me through the basics of the language and even helped me to break through the requirements so I could have a direction.  
  
In C#, using a Console App, I managed to create the game monster mayhem, with most of the rules and requirements, so I used it to develop an API in ASP.net. But then things started to get tricky and requiring much more knowledge, and I didn’t have enough time to go deep, and learn how to use those tools. So, with that I’ve decided to use the code which I’ve already done in C# used online tools to “convert” it to JavaScript, so I could be more familiar with the progress of the development.  
  
Things worked much better than my progress in those different languages, so I managed to get things done quickly and tailor the code to match the requirements. The hardest part was to implement the Concurrency into the code. For that I’ve read some tutorials online, videos, also used some “Chess” projects from GitHub so I could get a base of what should be done to have a concurrent system. (Links will be available on the references section).

The fact that I’ve tried to create Monster Mayhem in 3 different programming languages already made me proud of myself for the time and effort I’ve put into doing this project. The fact that my grad from the first CA was too low, incentive me even more to focus a lot in this CA2 to recover the max as I can.

I’m proud of my outcome with the Concurrency Class, unfortunately I’ve had to learn in the very last few weeks of this deadline how to build a game with real-time communication was a great experience.

# References

Developer, R.O.Node. js *et al.* (2023) *A simple guide to JavaScript concurrency in node.js*, *The Software House*. Available at: https://tsh.io/blog/simple-guide-concurrency-node-js/ (Accessed: 27 May 2024).

GeeksforGeeks (2022) *How to handle concurrency in node.js ?*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/how-to-handle-concurrency-in-node-js/ (Accessed: 28 May 2024).

*Client installation* (2024) *SocketIO RSS*. Available at: https://socket.io/docs/v4/client-installation/ (Accessed: 28 June 2024).

Socketio (no date) *Socketio/socket.io-client: Realtime application framework (client)*, *GitHub*. Available at: https://github.com/socketio/socket.io-client (Accessed: 28 May 2024).

Davymaish (no date) *Davymaish/2d-chess-game: This is a complete well automated multi-player chess game implemented in HTML5, CSS, and JavaScript.*, *GitHub*. Available at: https://github.com/davymaish/2d-chess-game (Accessed: 22 May 2024).

*CodePen: Online code editor and Front End Web Developer Community*. Available at: https://codepen.io/ (Accessed: 22 May 2024).

Yboodhan (no date) *Yboodhan/Ludo: 🎲 a multi-player strategy board game that takes a combination of luck and skill to win! uses Javascript, HTML, and CSS.*, *GitHub*. Available at: https://github.com/yboodhan/ludo (Accessed: 22 June 2024).