A page (at most) explaining the use of patterns in your design, or where you would use patterns if you had more time

In terms of the ***high cohesion*** pattern, I would have a master class always divided into subclasses that do each task. For example, when the game starts the tasks are divided by:

1. Showing the control buttons
2. Showing the board
3. Creating the variables for the new game
4. Choosing a first player
5. Starting the next turn

Because of this, its extremely easy to navigate around the project, and I am much more likely to pin-point problems when I am aware of what is happening.

As for the ***controller*** pattern, I use an overall facade controller, which is one system that runs every time a button is clicked. It uses a large if-else loop system, where certain commands cannot be called when an AI is currently taking a turn, preventing players from causing the game to possibly break.

Trying to have ***low coupling*** in this system could have been done better, most of the classes in the game do rely on each other, and also on the same global variables, so it is not very decoupled in that way. One issue I did have, was certain functions were only made for ‘Player 1’, because I originally planned on only having AI opponents. I had to go through many functions making them universally work, no matter who used them, to lower coupling. The way the game works, AI and human players all use very similar classes in many ways, just the way they take turns is different.