Which materials/key concepts from this course did you apply on the project?

One of the biggest key concepts from this course that we applied on this project was the Observer pattern. This helped us keep all our data in one place and update it and our views appropriately like we practiced on previous assignments. Aside from the main materials that are the MVC pattern and the Strategy Pattern, we also applied a lot of concepts from inheritance and interfaces in our first attempt at the project. Examples of how we used interfaces include when we created a Dark Mode and Light Mode version of our board which set a certain color scheme within methods that our BoardFormatter defines. Additionally, we created a ButtonListener that implements ActionListener in order to keep track of when the undo button was being used and making sure that the current player was being updated accurately. We also got to apply materials from Chapter 4 that deal with instances outside and inside the anonymous classes when trying to come up with a way to pass the results from the ActionListener class into creating a new board. Furthermore, we used concepts such as anonymous classes and GUI programming. Within our views, we had to implement anonymous classes of our Listeners. When we created a class for our MouseListener, we used the idea of saving it as an object so that we could pass it into a remover method when we no longer wanted the listeners to be active.

Which topics did you have to learn through self-study in order to complete the project?

In order to complete this project, our group practiced a mix of topics that we had learned in class as well as topics that we had to self-study. One of our first challenges was being able to switch the panels within a JFrame to move into the Tic Tac Toe board from the main menu. We learned that we had to remove the previous panel from the content pane before we set the content pane with another panel and had to revalidate it for it to show up. Another topic that was self-studied was the implementation of the Strategy Pattern. Since we did not get a chance to practice this pattern in class, this was our first time to utilize the pattern. The difficult part about the implementation of this pattern was having a defined concrete Strategy that uses a Strategy interface. Most previous implementations consisted of re- drawing the whole board within our supposed concrete pattern. At this point we knew that we were not implementing the pattern correctly. After a bit of research, we found out that the missing piece to our implementation was to have an aggregate of our strategy interface within our board. From there, our strategy instance can call upon its needed method to change the format of the board. The board itself would have a format() method where a concrete strategy completes the work that the class asks of it. Then, in our tester class, there would be two buttons, “light” and ”dark”, and each button would have an ActionListener that would create a concrete strategy that could be plugged into the format method of the bord.