1. **Which materials/key concepts from this course did you apply to the project?**

   For this project, we made different Use-cases, a UML Class diagram, and Sequence diagrams. We applied the Observer design pattern and Strategy design pattern on this project.

   Using the Observer pattern, we used key objects like Subject (BoardData class), Observers (CellButton and AnimatedCellButton classes), and the Controller class that calls the mutator of the model. We implemented the ActionListener Interface and used it as an anonymous class. Using the ActionListener, we implemented the actionPerformed() method which is invoked automatically when the user clicks on a component. While using the Strategy design pattern, we made a Strategy interface, Concrete strategies, and Context classes. We created the AnimatableButtonCreator interface as the Strategy interface. Then, we created NoAnimationButtonCreator and AnimatedButtonCreator classes, which implemented the AnimatableButtonCreator interface, as Concrete strategies. We also applied the concept of Inheritance and Polymorphism. We created CellButton and AnimatedCellButton classes, which extended the superclass GameView class. The class GameView, subclasses CellButton and AnimatedCellButton use the Strategy interface AnimatableButtonCreator to call the algorithm defined by NoAnimationButtonCreator and AnimatedButtonCreator concrete strategy classes. So, they are Context classes. We also made another interface called BoardType which is implemented by the superclass GameView and subclasses CellButton and AnimatedCellButton.

   For our convenience, we defined the GameView class as an abstract class. In this case, we reduced some repetitive codes and made our codes clearer. We also used JFrame, JPanel, JButton, JMenubar, JOptionPane, and JTextField. We put the undo button in JMenubar, so players can click to undo. We used JOptionPane to show “Game Over” when all the buttons are selected or “win” messages when one of the players wins. As long as one of these two things happens, this message frame will pop up. And at the very beginning, we use JTextFied to let players enter their name. The player’s name will show on the button they selected.

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

   We learned some of the concepts of Strategy design pattern through self-study. We learned how the Context class, Strategy interface, and concrete strategies work. We also learned about the concept of a Thread class that can be used to run multiple tasks simultaneously. The Thread class implements the Runnable interface and overrides it’s run() method which is used to perform an action for a thread. In order to complete this project, we also did extra research on the MVC design pattern and learned more about it.

Through this project, we have a deeper understanding of the MVC pattern. Now, we can use the Observer interface and Observable class skillfully.