**We both worked on this final documentation separately and combined into one.**

Our project is a barebones tic tac toe game. With basic features such as a player vs. player and computer vs player with a dialog displaying the winner. A previous version would automatically restart the game but there is a bug where the board is reset but the GUI is not. We’ve accomplished a lot with this project even though our project is basic we learned a lot about how to encapsulate and delegate classes to focus on specific problems allowing for both of us to work on the project and squash bugs without getting in each other’s way. I would say I was the so called ‘manager’ of the project. I created the file structure and determined how the classes would communicate with each other by defining the interfaces and general functions used for each class. We both worked on implementing the functions in the classes. Keegan originally implemented the GUI class, but it wasn’t very object oriented so we both broke down the GUI to better encapsulate the workings of the GUI from the engine. Keegan also helped implement the initial message asking if the player wants to play and if they clicked Yes it would play the game and if the user clicked no it would exit. The README shows where we ran into some issues along the way with this as well. Also, once this message was implemented Keegan had to build it so that a new game was started and also reset.

This project was very beneficial for learning. It exposed us to the world of JavaFX and using FXML files to build a GUI. While we did not implement the final project using FXML because I couldn’t quite figure out how to talk to the engine using FXML. This is the biggest obstacle we didn’t overcome, but the GUI we implemented had its kinks to work out that were very time consuming. Many of which were solved when we declared some global variables in the GUI class so the classes in the View package could access them. A problem we ran into after we got the GUI board to display correct user input, was adding buttons. The way we went about the GUI made it hard to add buttons to the screen, like start game and choose game type. Some lessons learned for me was how to use scene builder when creating an application, how to be more descriptive in naming variables, functions, classes, etc., and most importantly encapsulation, and interface segregation. I also learned that google is your friend if you ask it the right questions by understanding what you’re trying to implement.

Keegan: Before this class I had somewhat of an idea of what object-oriented programming was from previous classes I took but not an in-depth knowledge of this criteria. I started the class out on a rough start and continually struggled to get assignments completed with all the tests given passing. What this project has done is that it has helped me understand more about the aspect of object-oriented programming. Working with Tucker was a big help especially after I implemented the initial GUI. Learning how to encapsulate certain methods to the right classes really helped me grasp the whole concept of object-oriented programming. Some other ways I looked up how to do things was just like you said in class, is I open my browser and type in a question to Google and see where it takes me. Doing this helped me find some not so useful resources but some useful ones as well. There were a couple channels I found that helped me along the way, one channel called “newboston” and another called “kotlinschool” that I went to frequently whenever I didn’t understand a certain subject. Attending class every day was very beneficial as well as it allowed me to practice what subject you would be working on for that day.

