UI Development: The application's user interface is to be created using JavaFX. Scene Builder should be used to develop the interface based on FXML. If you choose to develop the interface using code to create the JavaFX objects rather than using FXML created by Scene Builder then you need to present a good reason for doing so in the final project documentation.

THIS REQUIREMENT WAS MET:

I used Scene Builder for all 3 of my FXML files. (AutoGameFXML.fxml, FXMLDocument.fxml, GameFXML.fxml)

Architecture: The application is to be built on the Model View Controller (MVC) architecture as shown in class. Required Elements: The following are elements that are required to be included in the application:

THIS REQUIREMENT WAS MET:

The VIEW's are FXML documents, the MODEL classes are GameModel and AutoGameModel, and the CONTROLLER classes are WarController, MenuController, and AutoWarController.

1. Object oriented elements that you write the code for:

a. Classes.

I created 10 classes for this project:

Card, Deck, PlayerHand, GameModel, AutoGameModel, AutoWarController, MenuController, Rank, Suit, WarController (Rank and Suit were enumerations, the controllers have controller in the name and the models have model in the name).

b. Subclasses.

PlayerHand is a subclass of the abstract class AbstractCardSet

c. At least one abstract class

AbstractCardSet is the abstract class I included.

d. At least one Interface

I included the interface GameInterface which was implemented in both GameModel and AutoGameModel

2. Code elements that you utilize:

a. One or more collection classes.

I used ArrayLists in PlayerHand, AutoGameModel, GameModel, Deck, and AbstractCardSet. (for quick proof look at Deck.java line 17)

b. Exception Handling.

Exception handling was used in the WarController's handleOpen, and handleSave functions. (WarController.handleOpen starting at line 194(try) ending at 209 (end of last

catch) WarController.handleSave starting at line 226(try) ending at 239(end of last catch)).

There is also exception handling involved with scene changes in the each controller (MenuController WarController, and AutoWarController).

3. The application must have a clearly defined model (as in the M in MVC). OO Final Project

Already explained below architecture section of this document.

4. The UI must utilize multiple scenes and at least one of the scenes will have the contents of the scene graph changed based on the application state.

I switch from the menu to each game scene and back. Switching scenes happens in the MenuController starting on line 57 with handleButtonAction1 and again with handleButtonAction2 on which starts on line 89. Once in a game you can return to the main menu by clicking File -> Main Menu. This is handled with the function goBackToPage1() in both AutoWarController (starting line 52) and WarController (starting line 183).

5. There must be a way to access "About" information that includes information about you and the application.

There is an About menu item on each scene. Clicking this will pop up an alert window with information about how to continue on the current page.

6. The application must save data and load data. The target for saving/loading data can be files, a network service, and/or a database.

You can save and load games, you cannot save and load AutoGames. This is handled in WarController as is mentioned in 2-b the part of this document covering exception handling.

Expectations:

1. The application is functional for a defined activity, task, and purpose. The goal is to develop a complete application not just a pile of code that doesn't serve a purpose.

This application serves the function of playing the card game war either manually (one card at a time) or automatically (determining the winner and displaying that information to the user)

2. The user interface is useable, organized, and understandable.

Makes sense to me and if you get confused there is information on each page just go to

Help -> About on the top menu bar.

3. The code is well-structured and logically organized.

I think I did a good job!

4. The application you build is not to be trivial in simply meeting the requirements set forth in this document. Yes, you are to meet the requirements but you are also to build an application that has a purpose and delivers functionality or capability. The requirements are parameters to be used in design and implementation of the application. They are not intended to be the end product.

It is a game it is not trivial.

5. You should design and build an application that you would be happy to show a prospective employer or client.

I am proud of my work and will be continuing to edit it and upload it to my Github.