# Planning

Possible Work Plan – **Change order if it makes more sense!**

Thurs Mar 18 – Create Test Plan

Fri Mar 19 – Design Classes

Mon Mar 22 – Board and Player

Tues Mar 23 – Console UI with validated game size and quit game

Wed Mar 24 – A Turn without validating it and undo move

Thurs Mar 25 – Validate a turn

Fri Mar 26 – Win or Tie checking

Mon Mar 29 – File reading, validation, and writing

Tues Mar 30 – Run full test plan and troubleshoot

Thurs Apr 1 – Make a drawing of the frame design

Tues Apr 6 – Make board show up in frame with New Game menu item

Wed Apr 7 - Make board update in frame instead of console

Thurs Apr 8 – Move all information or error messages to popups instead of console messages

Fri Apr 9 – Move Undo, Help, About, and Exit to menu items

Mon Apr 12 – Move Restore and Save Game to menu items

Tues Apr 13 – Move user move input to the frame

Wed Apr 14 – Run full test plan and troubleshoot

Parts of the program

**There must be ConnectNGame and ConnectNInterface classes**

Please use other classes (with associated objects in ConnectNGame and ConnectNInterface) to link together attributes and methods that share a common theme.

Board

* Has rows and columns
* Use a 2D array
* Should be initialized to E everywhere
* Should be updated after every turn

Player

* Get colour of each player
* Who starts first?
* Get player names
* Whose turn it is

Logic

* Check for a win
* Wins can happen in a row, column, or diagonal

Console Interface

* Greet the user!
* Give brief instructions
* Prompt for new game or continued game
* Prompt for board size and N
* Show board
* Prompt players for names
* Validate input for each move
* Prompt players for moves
* Give option to undo the most recent move
* Give option to save the game
* Give option to quit the game
* Ask player if they want to play again (reset game)

File Management

* Read in from a file
* Write out the game to a file in the appropriate format
* Validate file existence
* Validate format of file input

Validation

* Validate win conditions
* Validate board size, N-value
* Validate each move
* Validate if board is full

Turns

* Alternate between yellow and red
* Allow most recent move to be undone
* Make a move
* Allow quitting or saving
* Check for win conditions after every turn