# Requirement Trace Backs

Generally speaking, “this assignment consists of generating the graphical representation of a checkers board, and being able to specify initial piece positions” (from Assignment 1 specifications). The following will describe how the implementation of our classes reflects and accomplishes the prescribed requirements.

Note that changes to the design have been indicated with *New*, meaning we have implemented a new class previously inexistent, *New* *name*, meaning we have simply renamed the class for increased coherency, and *Revised*, meaning we have revised the class to manage its efficiency and tasks handled. Note that classes missing from this version have been removed since the last and are therefore irrelevant to the systems current traceability.

## Start\_Menu (*New*)

The Start\_Menu window is now the first one seen when launching the application as, originally, a player would simply be thrown into a standard game. However, with the option to save and load games, a more adequate solution was necessary to allow the user to choose which game method he/she prefers upon start-up. Usability is thus greatly enhanced. The page itself contains buttons including “Start Game” and “Load Game”. “Load Game” will launch a game previously saved (launches Load\_Game) whereas “Start Game” will give two game option: “Standard” or “Custom”. As is obvious by the names chosen, “Standard” will load a checkers board arranged with the pieces in a standard manner (ie. three rows of red and black pieces on opposite sides, launches Standard\_Mode) and “Custom” will give the player the opportunity to design a custom game, placing up to 12 pieces of each colour on any black tile (launches Custommode\_Load).

Even though Start\_Menu does not accomplish any of the requirements described in Assignment 2, it is imperative to the applications ease of use as it offers a simple user interface (UI) from which a player may select from many game modes.

## Play (*New*)

This form, adapted from the previous Form1\_Load, displays the general user interface. This includes, in broad terms, the game board, a title, a score board for each player and alphanumeric place holders as well as a menu strip located at the top.

The game board consists of light and dark squares (8 on the height and 8 on the width) from which a light square may be found in the bottom right corner. The board itself (modeled to imitate a wooden surface) lies on a coloured background image. The different colored tiles on the board are easy to differentiate, making piece recognition and placing rather simple and intuitive.

The board is also traced by a sequence of letters and numbers used to identify individual piece locations. The letters (placed on the top and bottom) start with A on the left and end with H on the right-most tile. Similarly, the numbers (found on the right and left sides of the board) start with 1 at the bottom tile and incrementally increase up to 8 at the top tile.

Therefore, Play manages to accomplish the “initial set up” of “an 8-by-8 checkers board with dark and light squares” in which “a light square” is found “in the bottom right corner” of the playing surface. Place naming conventions have also been met.

Originally, Play (or Form1\_Load) managed two independent modules which allowed their own playing experience. This, however, necessitated an enormous amount of code duplication. It was therefore decided that the Play form would be the one on which all gameplay would be controlled, eliminating otherwise necessary repetition and, hence, avoiding possible pitfalls such as duplicated bugs.

This module therefore imports game setups from Standard\_Mode, Custommode\_load and Load\_Game as well as containing all movement logic. The Play module allows a user to “start a game from a previously stored state”, “make moves” that include “mov[ing] pieces from one position to another” that include “mov[ing] a piece to another square; jump[ing] the opponent’s piece (so that piece is removed from the board); convert[ing] a piece to a “king” [and] mov[ing] kings in both directions (forwards and backwards)” (Assignment 2). This is all accomplished using a graphical user interface in which the user must simply click the piece he/she desires to move and then select the empty square they would like to move the piece to. The action will be completed providing the move is legal.

## Standard\_Mode (*New name* and *Revised*)

As was aforementioned, Start\_Menu included a button “Standard” under “Start Game”. Selecting this game option pushes the locations of the game pieces to the Play window, displaying a standard game piece arrangement. Hence, “the user [is] able to set up an initial position of pieces on the board by specifying […] the standard opening position”.

Upon pressing “Standard” in the start menu, the play window opens, replacing the initially viewed menu. It then places three rows of red pieces (white pieces on the Assignment specification diagram) on rows one to three, strictly positioned on dark tiles. Similarly, black pieces are placed on rows six to eight, once again, only on black tiles. A total of 12 pieces for each colour is placed on the board.

The initial game setup is therefore completely legal by default and obeys the initial setup requirements instilled by the assignment specifications.

## Custommode\_load (*New* *name*)

This class, as the name implies, allows the user to initiate a custom game in which pieces are placed at the user’s will. This mode is accessed through the start menu after selecting “Start Game” followed by “Custom”.

After selecting this mode of gameplay, a single red game piece appears at the bottom of Form1 accompanied by a “Complete Setup” button as well as a label that reads “Click on the left picture to change pieces”. At this moment, the red piece is selected and clicking on any black tile will place a red piece at that location. Clicking on the red piece outside of the board will circulate a red king piece. At this time, clicking on any black square will place a red king. A total of twelve combined king and standard red pieces may be placed. Clicking again on the selection piece will make a black piece appear followed by a black king piece, allowing the user to place black and black king pieces on the board respectively. Finally, pressing on the black king piece will show an empty space with the caption “Now removing. Click me to go to red”. This allows the user to remove pieces originally placed on the board. Simply clicking on any dark tile housing any piece will remove it from game play.

Tiles may be overwritten in custom mode’s set-up process. In other words, setting a piece of one color and then clicking the same space with a different piece will place the latter piece on the tile. Clicking multiple times on the same tile while in the same piece mode has no effect other than placing a piece on the first click. Clicking on any white space on the board during custom setup in any piece mode will display a pop-up window which reads “You cannot place a piece here” notifying the user that he/she has attempted an illegal procedure. Accepting this dialog box returns the player to the custom setup.

Once the player is content with the placement of all the game pieces, he/she may press the “Complete Setup” button at the bottom of the form. At this point, the Custommode\_load window disappears, giving way to the Play window which has imported the piece locations designated by the user while in custom set-up. The game is then ready to start.

Therefore, the player uses a graphical interface method of placing pieces as opposed to the much less intuitive and timely alternative method offered alongside in the Assignment specifications. Requirements such as “users shall be warned if the position is illegal” and “pieces must not be placed on illegal squares (white/light square)” as well as “a maximum of 12 [red] pieces and 12 black pieces may be placed on the board” are all met within this class’ interface. There is also “a way for the user to indicate that set up is complete” by using the “Complete Setup” button and commencing the game.

## Save\_Game (*New*)

The Save\_Game module allows a user to save a game in progress. At this point, any legal game play mode may be saved which includes standard and empty boards.

Saving a game may be accomplished through the menu strip located on the Play form under “Menu”. Selecting “Save Game” from these options will record piece locations in a file which may be retrieved whenever “Load Game” is executed. The state of the game in progress is therefore “saved within a file [which may] be resumed later” (Assignment 2).

Requirements are therefore met in order to save game progress at any moment during gameplay, allowing the user to record a games state to be retrieved and reloaded at a later date. Also note, since there is only one save slot at this time, that saving multiple times overwrites any older game state between the current game and past games.

## Load\_Game (*New*)

As opposed to an option within the Play menu strip, Load\_Game can be found as a game mode in the Start\_Menu form. Simply put, this allows Play to load piece locations saved previously using Save\_Game. Since there is, at the moment, only one storing state, Load\_Game loads the last-saved (most currently saved) game. This allows a user to play a game saved at an earlier date.

*Note that all requirements specified in the Assignments 1 and 2 have been met at this point and that the following modules are intended to offer supplementary functions in order to improve the user interface. They do not reflect requirements established in the Assignment documentation.*

## Reset (*Revised*)

This class is offered to accompany the above custom gameplay mode only. Selecting “Settings” on the menu strip of Custommode\_load displays the “Reset” option. This restores the board to the initial blank state of Custommode\_load, allowing the user to restart setting the piece locations from a “blank state”.

## GameTimer (*Revised*)

This class is simply the timer used during games. It is intended to allow the user to keep track his/her game play length. The timer restarts with new games and is not visible when a game mode is not selected. Note that the game timer was originally visible, but is now hidden upon further review.

## QuitGameToolStripMenuItem\_Click (*Revised*)

This class, activated by the user through the menu strip, can be found under “Menu”. Its purpose is to allow the user to quit the game without necessarily using the “close” button on the window bar. Selecting it activates a dialog box that reads “Are you sure you would like to quit?”. Clicking “Yes” will terminate the entire game, closing all windows, whereas selecting “No” will only close the message box and return the user to Play and its previous state. This feature has been reassigned exclusively to Custommode\_load for the time being.