UML changes explanation:

Ultimately there were minimal changes between milestone 3 and 4, the majority of the changes came from making several classes (Tile, GameModel, Player, Board and TileBag) implement the java module Serializable to implement the saving/loading functionality. Minimal changes were made to Board, GameFrame and GameModel to handle the undo/redo functions and custom board via XML files supported, as well as minor additions made to GameController to support undo/redo functionality. There were also minor changes to fix various bugs or glitches in the code, and ensure that the project was behaving as intended.

Data structure
Class:
Board:

- Data structures used:
    - 2D array: Tile[][] tempGrid
    - Explanation:
        - A scrabble board is always a fixed size, so a 2D array is appropriate as you'd never need to modify the size. It's also ideal as it allows for O(1) access to any cell. This data structure was not altered this milestone, but relates to an added method.
    - Operations:
        - removeTempTile(row, col) sets the space on the tempGrid back to Null
    - 2D array: Premium[][]
    - Explanation:
        - Since the board is always a fixed size, a 2D array is appropriate for listing the premium tiles.
    - Operations:
        - getPremium(row, col) returns the type of premium tile
        - setPremium(row, col, type) sets the tile as a specified premium tile

BoardLoader:

- Data structures used:
    - No data structures used
- Explanation:
    - Works by calling the function importXML that will draw data from the XML file and save it into usable board variables.

Dictionary:

- Data structures used:
    - No data structures were altered this milestone.
- Operations:
    - No operations were altered this milestone.

Player:

Data structures used:

No data structures were altered this milestone.

Operations:

No operations were altered this milestone.

PlacedTile:

Data structures used:

No data structures were altered this milestone.

Operations:

No operations were altered this milestone.

Tile:

Data structures used:

No data structures were altered this milestone.

Operations:

No operations were altered this milestone.

TileBag:

Data structures used:

No data structures were altered this milestone.

Operations:

No operations were altered this milestone.

GameFrame:

DataStructures used:

No data structures were altered this milestone.

Operations:

GameFrame() updated constructor to include menu item to save game
setUpPlayerOptions() updated this method to add undo and redo buttons

GameModel:

Data structures used:

No data structures were altered this milestone.

Operations:

saveGame(String) saves the current game
loadGame(String) loads the desired game
revertAction(ArrayList<PlacedTile>, ArrayList<PlacedTile>) performs the undo and redo action based on the order of the provided ArrayLists, and updates undoStack/redoStack accordingly

GameController:

Data structures used:

ArrayList<PlacedTile> undoStack

ArrayList<PlacedTile> redoStack

Explanation:

The ArrayLists for undoStack and redoStack are dynamic, which is important because the number of actions made in a turn will always vary. ArrayList also allows indexing using the getLast() method, which allows the ArrayList to be treated as a stack when handling undo and redo. Elements are removed using the getLast method so that the system doesn't have to loop through the entire stack each time an action is made, only when undo or redo is selected.

Operations:

actionPerformed(ActionEvent e) updated this method to handle undo and redo button events

handleUndo() handles the operations performed after an undo button click

handleRedo() handles the operations performed after a redo button click

nextPlayer() updated this method to clear undoStack and redoStack after a player's turn is over

placeTileTemporarily() updated this method to add to undoStack and clear redoStack when a player places a tile temporarily.