A lot of classes had changes for the undo/redo functionality, serialization/deserialization, and custom board selection features. We also added 2 classes.

New Classes

GameHistory

A new class called GameHistory was added to save the state of each turn for the undo/redo functionality. GameHistory uses multiple ArrayLists all for different reasons. One stores the tiles placed on the board during a turn. ArrayList was chosen because the number of tiles placed varies and we need O(1) access to iterate through them when we use undo. List<Tile> gainedTiles stores tiles that are drawn from the bag during a turn. ArrayList was chosen because it allows us to add tiles each time they're drawn and remove them in reverse order when using undo. List<Tile> swappedTiles stores the tiles that go back into the tile bag for a swap. ArrayList was chosen because any number of tiles can be swapped. List<int[]> newTilePositions stores the positions on the board of where the last turns tiles were placed. ArrayList was chosen because the number of newly placed letters can vary and we need to iterate through them to remove tiles for undo.

GameSerializer

A new class called GameSerializer was added to save and load the game. String constants (SAVE, FILE_EXTENSION) tells which folder the file will be saved and its extension. It uses ObjectOutputStream/ObjectInputStream, to write and read the objects to and from the files.

Modifications

ScrabbleModel

Multiple things were added to ScrabbleModel class for undo/redo and serialization. List<GameHistory> history stores every turn of the game. ArrayList was chosen because we need to be able to add new GameHistory objects multiple times. int pointInHistory is an index used to track the position in the history. It allows us to determine if undo/redo is possible and which GameHistory to access in O(1). transient List<ScrabbleView> views ensure that views are not serialized. transient Dictionary gameDictionary is used because the dictionary is reconstructed from the word file on load rather than being serialized. The addGameHistory() method removes every GameHistory that is past the current point in history, then adds hist to the end of GameHistory and moves the point in history forward 1 state. The undo() method undoes the changes stored in the current GameHistory and the redo() method Redoes the next play in the history

TileBag

transient Random random was added because we don't want Random objects serialized. A new one is created when deserialization occurs. addTileToStart(Tile tile) is a new method that adds a tile to the top of the bag. Used when undoing draws to maintain the order It keeps the tiles in the correct order for undo/redo.

Board

int boardselection is used to track which type of board is being used. setBoardselection(int boardtype) is used to update what board is being used and initializes the premium squares. initializePremiumSquares(int typeOfBoard) was modified to be able to input which type of board to initialize

Player, Tile, Board

Player, Tile and Board were modified for serialization. They now use Serializable to be able to save. private static final long serialVersionUID = 1L was added so that the versions are compatible for serialization/deserialization.

ScrabbleGUI

JButton undoButton, redoButton were added so that the user has a button to undo and redo. JButton was chosen to be consistent with other buttons. JMenu selectBoardMenu is a new menu with JMenuItems so the user can select the board type.

ScrabbleController

handleUndo() and handleRedo() were added to : Call model.undo() when their button is clicked

New Relationships

Composition: ScrabbleModel contains a List<GameHistory>
Dependency: GameSerializer depends on ScrabbleModel
Association: GameHistory references Tile objects
Serialization: Player, Tile, Board, TileBag, and ScrabbleModel implement Serializable interface