

## Java API View

## All Classes and Interfaces

All Classes and Interfaces	Interfaces	Classes	Enum Classes
Class	Description		
BlockIndex			
GameBoardView	The GameBoardView class visualizes the game board of blocks.		
GameStatusView	The GameStatusView class visualizes the game status as either in progress, current player's turn, or finished, either win or tie.		
Player			
RowBlockModel	The TicTacToeBlock class represents a given block in the game.		
RowGameApp			
RowGameController			
RowGameGUI	The RowGameGUI class is applying the Composite design pattern.		
RowGameModel			
UndoViewController			
View	The View interfaces supports updating the visualizations of the Model.		

## 10 Successful Tests (9 regression + 1 computer update test)

```

bill@billsair:~/spring2023/520/hw4/cs520-hw4/tictactoe$ ant test
Buildfile: /Users/bill/spring2023/520/hw4/cs520-hw4/tictactoe/build.xml

init:

compile:

compile.tests:
[javac] Compiling 1 source file to /Users/bill/spring2023/520/hw4/cs520-hw4/tictactoe/bin

test:
[echo] Running unit tests ...
[junit] Running TestExample
[junit] Testsuite: TestExample
[junit] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.446 sec
[junit] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.446 sec
[junit]
[junit] Testcase: testGameOverTie took 1.41 sec
[junit] Testcase: testGameOverWin took 0.005 sec
[junit] Testcase: testUndoAllowed took 0.003 sec
[junit] Testcase: testLegalMove took 0.004 sec
[junit] Testcase: testReset took 0.004 sec
[junit] Testcase: testComputerUpdate took 0.003 sec
[junit] Testcase: testUndoDisallowed took 0.003 sec
[junit] Testcase: testNewGame took 0.003 sec
[junit] Testcase: testNewBlockViolatesPrecondition took 0.003 sec
[junit] Testcase: testIllegalMove took 0.003 sec

BUILD SUCCESSFUL
Total time: 2 seconds

```

## Undo Breakpoint

```
85
86     /**
87      * Performs undo for the last move taken if the game is not
88      * in its initial configuration or has been finished.
89      */
90     public void undo() {
91         if (gameModel.movesLeft == 9 || gameModel.getFinalResult() != null) {
92             throw new UnsupportedOperationException(message: "Undo is currently disallowed.");
93         }
94
95         gameModel.undo();
96
97         gameView.update(gameModel);
98     }
```

## Variables before undo

```
✓ Local
  ✓ this: HumanRowGameController@25
    ✓ gameModel: RowGameModel@38
      ✓ blocksData: RowBlockModel[3][]@43
        ✓ 0: RowBlockModel[3]@61
          ✓ 0: RowBlockModel@65
            > contents: "X"
            > game: RowGameModel@38
              isLegalMove: false
            > 1: RowBlockModel@66
            > 2: RowBlockModel@67
          > 1: RowBlockModel[3]@62
          > 2: RowBlockModel[3]@63
          finalResult: null
        > historyOfMoves: Stack@44 size=1
          movesLeft: 8
          player: ☹ Player@45
      > gameView: RowGameGUI@40
```

Variables after undo

```
✓ Local
  ✓ this: HumanRowGameController@25
    ✓ gameModel: RowGameModel@38
      ✓ blocksData: RowBlockModel[3][]@43
        ✓ 0: RowBlockModel[3]@61
          ✓ 0: RowBlockModel@65
            > contents: ""
            > game: RowGameModel@38
              isLegalMove: true
            > 1: RowBlockModel@66
            > 2: RowBlockModel@67
          > 1: RowBlockModel[3]@62
          > 2: RowBlockModel[3]@63
          finalResult: null
          historyOfMoves: Stack@44 size=0
          movesLeft: 9
          player: 👁 Player@73
        > gameView: RowGameGUI@40
```