

Assignment 4

Generated by Doxygen 1.8.17

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Board< T > Class Template Reference	7
4.1.1 Detailed Description	7
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 Board()	8
4.1.3 Member Function Documentation	8
4.1.3.1 get()	8
4.1.3.2 getNumCol()	9
4.1.3.3 getNumRow()	9
4.1.3.4 set()	9
4.1.3.5 validPoint()	10
4.2 BoardController Class Reference	10
4.2.1 Constructor & Destructor Documentation	11
4.2.1.1 BoardController()	11
4.2.2 Member Function Documentation	11
4.2.2.1 closeViewStream()	11
4.2.2.2 get()	11
4.2.2.3 getInput()	12
4.2.2.4 modePrompt()	12
4.2.2.5 printMsg()	12
4.2.2.6 set()	12
4.2.2.7 updateBoard()	13
4.2.2.8 validateMoves()	13
4.3 BoardView Class Reference	13
4.3.1 Detailed Description	14
4.3.2 Member Function Documentation	14
4.3.2.1 getInput()	14
4.3.2.2 modePrompt()	14
4.3.2.3 printBoard()	14
4.3.2.4 printMsg()	15
4.4 Color Enum Reference	15
4.5 Dots Class Reference	15
4.6 GameEnd Class Reference	16
4.7 PointT Class Reference	16

4.7.1 Detailed Description	16
4.7.2 Constructor & Destructor Documentation	16
4.7.2.1 PointT()	16
4.7.3 Member Function Documentation	17
4.7.3.1 col()	17
4.7.3.2 row()	17
4.8 StrategyGameMode Class Reference	17
4.8.1 Detailed Description	18
4.9 TimedStrategy Class Reference	18
4.9.1 Detailed Description	18
4.10 TwoDotsBoard Class Reference	18
4.10.1 Detailed Description	19
4.10.2 Constructor & Destructor Documentation	19
4.10.2.1 TwoDotsBoard()	19
4.10.3 Member Function Documentation	19
4.10.3.1 updateBoard()	19
4.10.3.2 validateMoves()	20
5 File Documentation	21
5.1 src/BoardController.java File Reference	21
5.1.1 Detailed Description	21
5.2 src/Dots.java File Reference	21
5.2.1 Detailed Description	22
5.3 src/GameEnd.java File Reference	22
5.3.1 Detailed Description	22
5.4 src/PointT.java File Reference	22
5.4.1 Detailed Description	23
5.5 src/Strategy.java File Reference	23
5.5.1 Detailed Description	23
5.6 src/StrategyGameMode.java File Reference	23
5.6.1 Detailed Description	24
5.7 src/TimedStrategy.java File Reference	24
5.7.1 Detailed Description	24
5.8 src/TwoDotsBoard.java File Reference	24
5.8.1 Detailed Description	24
Index	25

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Board< T >	7
Board< Color >	7
TwoDotsBoard	18
BoardController	10
BoardView	13
Color	15
Dots	15
GameEnd	16
PointT	16
StrategyGameMode	17
TimedStrategy	18

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Board< T >	
Board provides an ADT to represent a generic 2D board parameterized over any type	7
BoardController	10
BoardView	
BoardView provides several access program to interact with a TwoDotsBoard object	13
Color	15
Dots	15
GameEnd	16
PointT	
PointT represents a single point in 2 dimensional space	16
StrategyGameMode	
Strategy is an interface for defining a family of game modes for Two Dots	17
TimedStrategy	18
TwoDotsBoard	
TwoDotsBoard provides an ADT to represent a TwoDots game board parameterized by the type Colors	18

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

src/ BoardController.java	
BoardController is a module used to interact with a TwoDotsBoard and a BoardView object . . .	21
src/ Dots.java	
Dots contains the main client code to start a game of TwoDots	21
src/ GameEnd.java	
GameEnd contains routines to create a count down timer and execute a function when this timer is out of time	22
src/ PointT.java	
PointT is a module used to represent a point in 2D space	22
src/ Strategy.java	
Strategy is an interface for defining a family of game modes for Two Dots	23
src/ StrategyGameMode.java	
StrategyGameMode defines a generic algorithm for playing a Strategy/game mode for TwoDots	23
src/ TimedStrategy.java	
TimedStrategy is the timed game mode for TwoDots	24
src/ TwoDotsBoard.java	
TwoDotsBoard extends Board and is parameterized with type Colors	24

Chapter 4

Class Documentation

4.1 Board< T > Class Template Reference

[Board](#) provides an ADT to represent a generic 2D board parameterized over any type.

Collaboration diagram for Board< T >:

Public Member Functions

- [Board](#) (int row, int col)
Constructor method for [Board](#).
- void [set](#) ([PointT](#) p, T v)
setter method for [Board](#), get a value at a given point
- T [get](#) ([PointT](#) p)
getter method for [Board](#), get a value at a given point
- int [getNumRow](#) ()
getter method for the number of rows in the current [Board](#) object
- int [getNumCol](#) ()
getter method for the number of columns in the current [Board](#) object

Protected Member Functions

- boolean [validPoint](#) ([PointT](#) p)
protected helper method to validate a [PointT](#) object. visible to the current and a child class only

Protected Attributes

- ArrayList< ArrayList< T > > **s**
- int **n_row**
- int **n_col**

4.1.1 Detailed Description

[Board](#) provides an ADT to represent a generic 2D board parameterized over any type.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Board()

```
Board< T >.Board (
    int row,
    int col )
```

Constructor method for [Board](#).

Parameters

<i>row</i>	the number of rows desired in the board
<i>col</i>	the number of columns desired in the board

the Boards cells are not initialized to anything, only the corresponding number of rows are added

Exceptions

<i>IllegalArgumentException</i>	if the parameter row or parameter col is less than or equal to 0
---------------------------------	--

4.1.3 Member Function Documentation

4.1.3.1 get()

```
T Board< T >.get (
    PointT p )
```

getter method for [Board](#), get a value at a given point

Parameters

<i>p</i>	PointT object indicating the position to get
----------	--

Returns

the value at point p

Exceptions

<i>IndexOutOfBoundsException</i>	if PointT object lies outside of the Board , i.e the row or column lie outside of the Boards dimensions
----------------------------------	---

4.1.3.2 getNumCol()

```
int Board< T >.getNumCol ( )
```

getter method for the number of columns in the current Board object

Returns

the number of columns in the current Board object

4.1.3.3 getNumRow()

```
int Board< T >.getNumRow ( )
```

getter method for the number of rows in the current Board object

Returns

the number of rows in the current Board

4.1.3.4 set()

```
void Board< T >.set (
    PointT p,
    T v )
```

setter method for Board, get a value at a given point

Parameters

<i>p</i>	PointT object indicating the position to get
----------	--

Returns

the value at point p

Exceptions

<i>IndexOutOfBoundsException</i>	if PointT object lies outside of the Board, i.e the row or column lie outside of the Boards dimensions
----------------------------------	--

4.1.3.5 validPoint()

```
boolean Board< T >.validPoint (
    PointT p ) [protected]
```

protected helper method to validate a [PointT](#) object. visible to the current and a child class only

Parameters

<i>p</i>	the PointT object to validate
----------	---

Returns

boolean indicating if the column number is valid (i.e lies in the 2D sequence bounds)

The documentation for this class was generated from the following file:

- src/Board.java

4.2 BoardController Class Reference

Public Member Functions

- [BoardController](#) ([TwoDotsBoard](#) model, [BoardView](#) view)
constructor for [BoardController](#)
- [Color](#) get ([PointT](#) p)
get the color at a specific point for a [TwoDotsBoard](#) object
- void set ([PointT](#) p, [Color](#) c)
set a color at a specific point for a [TwoDotsBoard](#) object
- boolean validateMoves ([BoardMoves](#) moves)
validate a sequence of input moves for a [TwoDotsBoard](#)
- void updateBoard ([BoardMoves](#) moves)
update a [TwoDotsBoard](#) after eliminating a sequence of dots
- void updateView ()
print out to standard output the current state of a [TwoDotsBoard](#)
- void printMsg (String msg)
print out to standard input a string
- Strategy modePrompt ()
get a playable game mode of choice from the user through the standard input via a prompt
- void closeViewStream ()
close the input stream from the standard input
- [BoardMoves](#) getInput ()
get a input sequence of dots to eliminate on a [TwoDotsBoard](#) from the user via the standard input

4.2.1 Constructor & Destructor Documentation

4.2.1.1 BoardController()

```
BoardController.BoardController (
    TwoDotsBoard model,
    BoardView view )
```

constructor for [BoardController](#)

Parameters

<i>model</i>	the TwoDotsBoard object to control, interact with
<i>view</i>	the view object to interact with

4.2.2 Member Function Documentation

4.2.2.1 closeViewStream()

```
void BoardController.closeViewStream ( )
```

close the input stream from the standard input

Note: this is not specific to a specific instance of a [BoardController](#) object and will close the stream globally, use carefully

4.2.2.2 get()

```
Color BoardController.get (
    PointT p )
```

get the color at a specific point for a [TwoDotsBoard](#) object

Parameters

<i>p</i>	The point to get the color at
----------	-------------------------------

Returns

the [Color](#) at point p

4.2.2.3 getInput()

```
BoardMoves BoardController.getInput ( )
```

get a input sequence of dots to eliminate on a [TwoDotsBoard](#) from the user via the standard input

Returns

a sequence containing the points indicated by the user

4.2.2.4 modePrompt()

```
Strategy BoardController.modePrompt ( )
```

get a playable game mode of choice from the user through the standard input via a prompt

see [modePrompt\(\)](#) in BoardView.java for more information

Returns

the game mode the user wants to play

4.2.2.5 printMsg()

```
void BoardController.printMsg (
    String msg )
```

print out to standard input a string

Parameters

<i>msg</i>	the string to print out
------------	-------------------------

4.2.2.6 set()

```
void BoardController.set (
    PointT p,
    Color c )
```

set a color at a specific point for a [TwoDotsBoard](#) object

Parameters

<i>p</i>	The point to set the color at
<i>c</i>	the color to set

4.2.2.7 updateBoard()

```
void BoardController.updateBoard (
    BoardMoves moves )
```

update a [TwoDotsBoard](#) after eliminating a sequence of dots

See updateBoard in [TwoDotsBoard](#) for more information

Parameters

<i>moves</i>	containing the sequence of dots to eliminate
--------------	--

4.2.2.8 validateMoves()

```
boolean BoardController.validateMoves (
    BoardMoves moves )
```

validate a sequence of input moves for a [TwoDotsBoard](#)

see validateMoves in [TwoDotsBoard](#) for more information

Parameters

<i>moves</i>	the sequence of moves to validate
--------------	-----------------------------------

Returns

boolean indicating is the given sequence is valid

The documentation for this class was generated from the following file:

- [src/BoardController.java](#)

4.3 BoardView Class Reference

[BoardView](#) provides several access program to interact with a [TwoDotsBoard](#) object.

Collaboration diagram for BoardView:

Public Member Functions

- void `printBoard` (`TwoDotsBoard` board)
print the contents of a `TwoDotsBoard` to standard output in a tabular manner
- Strategy `modePrompt` ()
get user input for which game mode to play
- BoardMoves `getInput` ()
get user input for which dots to eliminate on the `TwoDotsBoard`, sequence of `PointT` objects
- void `closeStream` ()
close the scanner stream
- void `printMsg` (String msg)
print out a string to the standard output

4.3.1 Detailed Description

`BoardView` provides several access program to interact with a `TwoDotsBoard` object.

4.3.2 Member Function Documentation

4.3.2.1 `getInput()`

```
BoardMoves BoardView.getInput ( )
```

get user input for which dots to eliminate on the `TwoDotsBoard`, sequence of `PointT` objects
will continuously ask for input until user enters correct format

Returns

BoardMoves object containing the dots to eliminate

4.3.2.2 `modePrompt()`

```
Strategy BoardView.modePrompt ( )
```

get user input for which game mode to play

Returns

the game mode the user wants to play

4.3.2.3 `printBoard()`

```
void BoardView.printBoard (
    TwoDotsBoard board )
```

print the contents of a `TwoDotsBoard` to standard output in a tabular manner

Parameters

<i>the</i>	board to print out
------------	--------------------

4.3.2.4 printMsg()

```
void BoardView.printMsg (  
    String msg )
```

print out a string to the standard output

Parameters

<i>msg</i>	the message to print
------------	----------------------

The documentation for this class was generated from the following file:

- [src/BoardView.java](#)

4.4 Color Enum Reference

Static Public Member Functions

- static [Color](#) randomColor ()

Public Attributes

- **R**
- **G**
- **B**
- **P**
- **O**

The documentation for this enum was generated from the following file:

- [src/Color.java](#)

4.5 Dots Class Reference

Static Public Member Functions

- static void **main** (String[] args)

The documentation for this class was generated from the following file:

- [src/Dots.java](#)

4.6 GameEnd Class Reference

Collaboration diagram for GameEnd:

Classes

- class **End**

Public Member Functions

- **GameEnd** (int time)

The documentation for this class was generated from the following file:

- src/[GameEnd.java](#)

4.7 PointT Class Reference

[PointT](#) represents a single point in 2 dimensional space.

Public Member Functions

- [PointT](#) (int [row](#), int [col](#))
constructor method for [PointT](#)
- int [row](#) ()
getter method for the row value the point is on
- int [col](#) ()
getter method for the column value the point is on

4.7.1 Detailed Description

[PointT](#) represents a single point in 2 dimensional space.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 PointT()

```
PointT.PointT (  
    int row,  
    int col )
```

constructor method for [PointT](#)

Parameters

<i>row</i>	the row number to create the point at
<i>col</i>	the column number to create the point at

4.7.3 Member Function Documentation

4.7.3.1 col()

```
int PointT.col ( )
```

getter method for the column value the point is on

Returns

the column the point is on

4.7.3.2 row()

```
int PointT.row ( )
```

getter method for the row value the point is on

Returns

the row the point is on

The documentation for this class was generated from the following file:

- [src/PointT.java](#)

4.8 StrategyGameMode Class Reference

Strategy is an interface for defining a family of game modes for Two [Dots](#).

Inheritance diagram for StrategyGameMode:

Collaboration diagram for StrategyGameMode:

Public Member Functions

- void **play** ([TwoDotsBoard](#) b)

Protected Attributes

- [BoardView](#) **v**
- [BoardController](#) **c**
- BoardMoves **moves**

4.8.1 Detailed Description

Strategy is an interface for defining a family of game modes for Two [Dots](#).

implements the Strategy interface

The documentation for this class was generated from the following file:

- [src/StrategyGameMode.java](#)

4.9 TimedStrategy Class Reference

Inheritance diagram for TimedStrategy:

Collaboration diagram for TimedStrategy:

Additional Inherited Members

4.9.1 Detailed Description

extends [StrategyGameMode](#) to implement a customizable game play

The documentation for this class was generated from the following file:

- [src/TimedStrategy.java](#)

4.10 TwoDotsBoard Class Reference

[TwoDotsBoard](#) provides an ADT to represent a TwoDots game board parameterized by the type Colors.

Inheritance diagram for TwoDotsBoard:

Collaboration diagram for TwoDotsBoard:

Public Member Functions

- [TwoDotsBoard](#) (int row, int col)
Constructor method for [TwoDotsBoard](#).
- boolean [validateMoves](#) (BoardMoves moves)
check if a given BoardMoves sequence is valid set of moves on the two dots board
- void [updateBoard](#) (BoardMoves moves)
setter method to update the board after

Additional Inherited Members

4.10.1 Detailed Description

[TwoDotsBoard](#) provides an ADT to represent a TwoDots game board parameterized by the type Colors.

extends from [Board](#) class

4.10.2 Constructor & Destructor Documentation

4.10.2.1 TwoDotsBoard()

```
TwoDotsBoard::TwoDotsBoard (
    int row,
    int col )
```

Constructor method for [TwoDotsBoard](#).

Parameters

<i>row</i>	the number of rows desired in the board
<i>col</i>	the number of columns desired in the board

The board is initialized to random colors

Exceptions

<i>IllegalArgumentException</i>	if the parameter row or parameter col is less than or equal to 0
---------------------------------	--

4.10.3 Member Function Documentation

4.10.3.1 updateBoard()

```
void TwoDotsBoard::updateBoard (
    BoardMoves moves )
```

setter method to update the board after

set new random values after eliminating the target points

Parameters

<i>moves</i>	sequence of BoardMoves containing the cells on the Board to remove
--------------	--

4.10.3.2 validateMoves()

```
boolean TwoDotsBoard.validateMoves (
    BoardMoves moves )
```

check if a given BoardMoves sequence is valid set of moves on the two dots board

a sequence of size less than or equal to 1 is not valid, a sequence with the same dot visited twice is not valid, a sequence with adjacent dots that do not have the same color are also not valid and if a point is not on the board

Parameters

<i>moves</i>	sequence of points on the board to validate
--------------	---

Returns

boolean indicating the result

The documentation for this class was generated from the following file:

- src/[TwoDotsBoard.java](#)

Chapter 5

File Documentation

5.1 src/BoardController.java File Reference

[BoardController](#) is a module used to interact with a [TwoDotsBoard](#) and a [BoardView](#) object.

Classes

- class [BoardController](#)

5.1.1 Detailed Description

[BoardController](#) is a module used to interact with a [TwoDotsBoard](#) and a [BoardView](#) object.

Author

Shazil Arif

Date

April 2nd 2020

5.2 src/Dots.java File Reference

[Dots](#) contains the main client code to start a game of TwoDots.

Classes

- class [Dots](#)

5.2.1 Detailed Description

[Dots](#) contains the main client code to start a game of TwoDots.

Author

Shazil Arif

Date

April 1st 2020

5.3 src/GameEnd.java File Reference

[GameEnd](#) contains routines to create a count down timer and execute a function when this timer is out of time.

Classes

- class [GameEnd](#)
- class [GameEnd.End](#)

5.3.1 Detailed Description

[GameEnd](#) contains routines to create a count down timer and execute a function when this timer is out of time.

Author

Shazil Arif

Date

April 1st 2020

5.4 src/PointT.java File Reference

[PointT](#) is a module used to represent a point in 2D space.

Classes

- class [PointT](#)
[PointT](#) represents a single point in 2 dimensional space.

5.4.1 Detailed Description

[PointT](#) is a module used to represent a point in 2D space.

Author

Shazil Arif

Date

April 1st 2020

5.5 src/Strategy.java File Reference

Strategy is an interface for defining a family of game modes for Two [Dots](#).

Classes

- interface **Strategy**

Strategy is an interface for defining a family of game modes for Two [Dots](#).

5.5.1 Detailed Description

Strategy is an interface for defining a family of game modes for Two [Dots](#).

Author

Shazil Arif

Date

April 1st 2020

5.6 src/StrategyGameMode.java File Reference

[StrategyGameMode](#) defines a generic algorithm for playing a Strategy/game mode for TwoDots.

Classes

- class [StrategyGameMode](#)

Strategy is an interface for defining a family of game modes for Two [Dots](#).

5.6.1 Detailed Description

[StrategyGameMode](#) defines a generic algorithm for playing a Strategy/game mode for TwoDots.

Author

Shazil Arif

Date

April 2nd 2020

5.7 [src/TimedStrategy.java](#) File Reference

[TimedStrategy](#) is the timed game mode for TwoDots.

Classes

- class [TimedStrategy](#)

5.7.1 Detailed Description

[TimedStrategy](#) is the timed game mode for TwoDots.

Author

Shazil Arif

Date

April 2nd 2020

5.8 [src/TwoDotsBoard.java](#) File Reference

[TwoDotsBoard](#) extends [Board](#) and is parameterized with type Colors.

Classes

- class [TwoDotsBoard](#)
[TwoDotsBoard](#) provides an ADT to represent a TwoDots game board parameterized by the type Colors.

5.8.1 Detailed Description

[TwoDotsBoard](#) extends [Board](#) and is parameterized with type Colors.

Author

Shazil Arif

Date

April 1st 2020

Index

- Board
 - Board< T >, 8
- Board< T >, 7
 - Board, 8
 - get, 8
 - getNumCol, 9
 - getNumRow, 9
 - set, 9
 - validPoint, 10
- BoardController, 10
 - BoardController, 11
 - closeViewStream, 11
 - get, 11
 - getInput, 11
 - modePrompt, 12
 - printMsg, 12
 - set, 12
 - updateBoard, 13
 - validateMoves, 13
- BoardView, 13
 - getInput, 14
 - modePrompt, 14
 - printBoard, 14
 - printMsg, 15
- closeViewStream
 - BoardController, 11
- col
 - PointT, 17
- Color, 15
- Dots, 15
- GameEnd, 16
- get
 - Board< T >, 8
 - BoardController, 11
- getInput
 - BoardController, 11
 - BoardView, 14
- getNumCol
 - Board< T >, 9
- getNumRow
 - Board< T >, 9
- modePrompt
 - BoardController, 12
 - BoardView, 14
- PointT, 16
 - col, 17
 - PointT, 16
 - row, 17
- printBoard
 - BoardView, 14
- printMsg
 - BoardController, 12
 - BoardView, 15
- row
 - PointT, 17
- set
 - Board< T >, 9
 - BoardController, 12
- src/BoardController.java, 21
- src/Dots.java, 21
- src/GameEnd.java, 22
- src/PointT.java, 22
- src/Strategy.java, 23
- src/StrategyGameMode.java, 23
- src/TimedStrategy.java, 24
- src/TwoDotsBoard.java, 24
- StrategyGameMode, 17
- TimedStrategy, 18
- TwoDotsBoard, 18
 - TwoDotsBoard, 19
 - updateBoard, 19
 - validateMoves, 20
- updateBoard
 - BoardController, 13
 - TwoDotsBoard, 19
- validateMoves
 - BoardController, 13
 - TwoDotsBoard, 20
- validPoint
 - Board< T >, 10