Assignment 4: Game Of Dots

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 AllTests Class Reference

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

- src/AllTests.java

3.2 src.CellT Enum Reference

This class represents CellT.

Static Public Member Functions

- static CellT getRandomCell ()
  method that returns a random CellT type

Public Attributes

- **R**
- **O**
- **G**

3.2.1 Detailed Description

This class represents CellT.

This is an enum class for the various CellT types

3.2.2 Member Function Documentation

3.2.2.1 getRandomCell()

```
static CellT src.CellT.getRandomCell ( )  [static]
```

method that returns a random CellT type

Method that returns either of the four CellT colors

Returns

Returns a random CellT value

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

- src/CellT.java

3.3 src.Controller Class Reference

This class outputs the controller to update model and view.

Public Member Functions

- Controller (GameBoardT gameboard, View view)
  Constructor that that initializes for the controller.
- CellT [ ][ ] get_game_boardT ()
  Getter method used to retrieve boardT grid.
- ArrayList get_game_obj ()
  Getter method used to retrieve objectives.
- int get_game_score ()
  Getter method used to retrieve score.
- int get_game_moves ()
  Getter method used to retrieve number of moves.
- void execute_move (int[ ][ ] path)
  Controller Method that updates the boardT after a valid move.
- void updateView ()
  Controller method that updates the view output for the user.
- boolean game_over ()
  Checks if game is over when ran out of available moves.
- boolean is_win ()
  Checks if user has won the game.

3.3.1 Detailed Description

This class outputs the controller to update model and view.

Controller of the MVC design pattern

3.3.2 Constructor & Destructor Documentation

3.3.2.1 Controller()

```
src.Controller.Controller (
            GameBoardT gameboard,
            View view )
```

Constructor that that initializes for the controller.

Initializes the model(gameboard) and view modules

Parameters

gameboard	represents a GameboardT object
view	represents a View object

3.3.3 Member Function Documentation

3.3.3.1 execute_move()

```
void src.Controller.execute_move (
            int path[][] )
```

Controller Method that updates the boardT after a valid move.

Mutator method that updates the boardT with a valid path from user

Parameters

path	represents the 2D integer array of the path chosen by the client

Exceptions

NullPointerException,IndexOutOfBoundsException,IllegalArgumentException	

3.3.3.2 game_over()

```
boolean src.Controller.game_over ( )
```

Checks if game is over when ran out of available moves.

Outputs a boolean signifying if available moves is < = 0

Returns

returns True if available moves is less than or equal to 0

3.3.3.3 get_game_boardT()

```
CellT [][] src.Controller.get_game_boardT ( )
```

Getter method used to retrieve boardT grid.

getter method to return the boardT state variable

Returns

Returns a 2D array of CellT representing the boardT

3.3.3.4 get_game_moves()

```
int src.Controller.get_game_moves ( )
```

Getter method used to retrieve number of moves.

Gets the number of moves remaining anytime during the game

Returns

Returns an integer containing the number of moves remaining

3.3.3.5 get_game_obj()

```
ArrayList src.Controller.get_game_obj ( )
```

Getter method used to retrieve objectives.

getter method to return the objectives state variable

Returns

Returns an arraylist containing objective parameters

3.3.3.6 get_game_score()

```
int src.Controller.get_game_score ( )
```

Getter method used to retrieve score.

getter method to return the score state variable

Returns

Returns an integer containing the score

3.3.3.7 is_win()

```
boolean src.Controller.is_win ( )
```

Checks if user has won the game.

Outputs a boolean if user has met the objective

Returns

returns True if objective has reached or gone below 0

3.3.3.8 updateView()

```
void src.Controller.updateView ( )
```

Controller method that updates the view output for the user.

Outputs the state variables for the game by printing to the console

Exceptions

NullPointerException,IndexOutOfBoundsException,IllegalArgumentException	

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

- src/Controller.java

3.4 src.GameBoardT Class Reference

class represents the GameBoardT module to play Dots Game

Public Member Functions

- GameBoardT ()

  Constructor that initializes the GameBoardT object.
- void initialize_board ()

  Initializes the GameBoard for the state variables.
- CellT [ ][ ] get_boardT ()

  Getter method used to retrieve boardT grid.
- ArrayList get_obj ()

  Getter method used to retrieve objectives.
- int get_score ()

  Getter method used to retrieve score.
- int get_moves ()

  Getter method used to retrieve number of moves.
- void move (int[ ][ ]path)

  Method that updates the boardT after a valid move.
- boolean game_over ()

  Checks if game is over when ran out of available moves.
- boolean is_win ()

  Checks if user has won the game.

3.4.1 Detailed Description

class represents the GameBoardT module to play Dots Game

The model of MVC with various operations on a 6X6 grid

3.4.2 Constructor & Destructor Documentation

3.4.2.1 GameBoardT()

```
src.GameBoardT.GameBoardT ( )
```

Constructor that initializes the GameBoardT object.

GameBoardT constructor takes no parameters to create GameBoardT object

3.4.3 Member Function Documentation

3.4.3.1 game_over()

```
boolean src.GameBoardT.game_over ( )
```

Checks if game is over when ran out of available moves.

Outputs a boolean signifying if available moves is $<$ = 0

Returns

returns True if available moves is less than or equal to 0

3.4.3.2 get_boardT()

```
CellT [][] src.GameBoardT.get_boardT ( )
```

Getter method used to retrieve boardT grid.

getter method to return the boardT state variable

Returns

Returns a 2D array of CellT representing the boardT

3.4.3.3 get_moves()

```
int src.GameBoardT.get_moves ( )
```

Getter method used to retrieve number of moves.

Gets the number of moves remaining anytime during the game

Returns

Returns an integer containing the number of moves remaining

3.4.3.4 get_obj()

```
ArrayList src.GameBoardT.get_obj ( )
```

Getter method used to retrieve objectives.

getter method to return the objectives state variable

Returns

Returns an arraylist containing objective parameters

3.4.3.5 get_score()

```
int src.GameBoardT.get_score ( )
```

Getter method used to retrieve score.

getter method to return the score state variable

Returns

Returns an integer containing the score

3.4.3.6 initialize_board()

```
void src.GameBoardT.initialize_board ( )
```

Initializes the GameBoard for the state variables.

Takes no parameters. Assigns the default properties for the board

3.4.3.7 is_win()

```
boolean src.GameBoardT.is_win ( )
```

Checks if user has won the game.

Outputs a boolean if user has met the objective

Returns

returns True if objective has reached or gone below 0

3.4.3.8 move()

```
void src.GameBoardT.move (
            intpath [][] )
```

Method that updates the boardT after a valid move.

Mutator method that updates the boardT with a valid path from user

Parameters

path	represents the 2D integer array of the path chosen by the client

Exceptions

NullPointerException,IndexOutOfBoundsException,IllegalArgumentException	

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

- src/GameBoardT.java

3.5 src.TestCellT Class Reference

Public Member Functions

- void **setUp** () throws Exception
- void **tearDown** () throws Exception
- void **test_getRandomCell** ()

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

- src/TestCellT.java

3.6 src.TestController Class Reference

Public Member Functions

- void **setUp** () throws Exception
- void **tearDown** () throws Exception
- void **test_get_boardT** ()
- void **test_get_obj** ()
- void **test_get_score** ()
- void **test_get_moves** ()
- void **test_game_over** ()
- void **test_is_win** ()
- void **test_move** ()

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

- src/TestController.java

3.7 src.TestGameBoardT Class Reference

Public Member Functions

- void **setUp** () throws Exception
- void **tearDown** () throws Exception
- void **test_get_boardT** ()
- void **test_get_obj** ()
- void **test_get_score** ()
- void **test_get_moves** ()
- void **test_game_over** ()
- void **test_is_win** ()
- void **test_move** ()

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

- src/TestGameBoardT.java

3.8 src.View Class Reference

This class outputs the board and game details to the user.

Public Member Functions

- void print_board (CellT[ ][ ] boardT)
  Method used to show the board to the user.
- void print_info (CellT objective_type, int objective_goal, int score, int moves)
  Method used to show the users status in the game.

3.8.1 Detailed Description

This class outputs the board and game details to the user.

This is the view portion of the MVC design pattern

3.8.2 Member Function Documentation

3.8.2.1 print_board()

```
void src.View.print_board (
            CellT boardT[][] )
```

Method used to show the board to the user.

Prints the board to the console to view

Parameters

boardT	is a 2D sequence of CellT that represents the game board

3.8.2.2 print_info()

```
void src.View.print_info (
            CellT objective_type,
            int objective_goal,
            int score,
            int moves )
```

Method used to show the users status in the game.

Prints the current objectives, moves, and score to the console

Parameters

objective_type	represents the CellT objective
objective_goal	represents the target goal for CellT objective
score	represents the running total of the users points
moves	represents total available moves for the user

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

- src/View.java

Chapter 4

File Documentation

4.1 src/CellT.java File Reference

Enum Java class for CellT.

Classes

- enum src.CellT
  This class represents CellT.

4.1.1 Detailed Description

Enum Java class for CellT.

Author

Anando Zaman

Date

March 25,2020

Used for the exported LandTypes

4.2 src/Controller.java File Reference

Controller class for model and view modules.

Classes

- class src.Controller
  This class outputs the controller to update model and view.

4.2.1 Detailed Description

Controller class for model and view modules.

Author

Anando Zaman

Date

March 25,2020

Used to control/update model(GameBoardT) and view modules

4.3 src/GameBoardT.java File Reference

GameBoardT ADT module that uses CellT module.

Classes

- class src.GameBoardT
  class represents the GameBoardT module to play Dots Game

4.3.1 Detailed Description

GameBoardT ADT module that uses CellT module.

Author

Anando Zaman

Date

March 25,2020

Model of MVC for playing the Dots game on a 6X6 GRID

4.4 src/TestCellT.java File Reference

Test Driver for CellT module.

Classes

- class src.TestCellT

4.4.1 Detailed Description

Test Driver for CellT module.

Author

Anando Zaman

Date

March 25,2020

4.5 src/TestGameBoardT.java File Reference

Test Driver for GameBoardT module.

Classes

- class src.TestGameBoardT

4.5.1 Detailed Description

Test Driver for GameBoardT module.

Test Driver for Controller module.

Author

Anando Zaman

Date

March 25,2020

4.6 src/View.java File Reference

View module that uses CellT and GameBoardT modules.

Classes

- class src.View
  This class outputs the board and game details to the user.

4.6.1 Detailed Description

View module that uses CellT and GameBoardT modules.

Author

Anando Zaman

Date

March 25,2020

View of MVC for outputting results to the user

Index