Assignment 4

Generated by Doxygen 1.9.1

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 Package src.controller

Classes

- class BoardManager

  A library containing useful functions for manipulating BoardT objects.
- class Controller

  An abstract object that represents a controller used to make moves and check the state of the 2048 game.

4.1.1 Detailed Description

Author: Saruggan Thiruchelvan (thirus6) Revised: April 10, 2021

Description: BoardManager Module (Library)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 11, 2021

Description: Controller Module (Abstract Object)

4.2 Package src.model

Classes

- enum DirectionT

  An enumerated type that represents the direction of a given move in the game (i.e. the arrow key directions).
- class BoardT

  An ADT that implements the board for the 2048 game using a matrix (2-dimensional array) of integer values to represent the state of the board.
- class ScoreT

  An abstract object that maintains the score and high score of the game.

4.2.1 Detailed Description

Author: Saruggan Thiruchelvan (thirus6) Revised: April 10, 2021

Description: DirectionT Module (Enumerated Type)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 10, 2021

Description: BoardT Module (Abstract Data Type)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 10, 2021

Description: ScoreT Module (Abstract Object)

4.3 Package src.view

Classes

- class View

  An abstract object for displaying the the 2048 game using a graphical user interface (GUI).
- class BoardUI

  An abstract object to display and update the view of the current state of the 2048 game board from the Controller module.
- interface ComponentUI

  An interface for UI components.
- class MessageUI

  An abstract object to display and update the view of the current state the 2048 game from the Controller module with respect to whether the player has achieved the 2048 tile or if the game is over.
- class ScoreUI

  An abstract object to display and update the view of the current score and high score of the 2048 game from the ScoreT module.
- class TileUI

  An abstract data type to display and update the view of an individual tile in the 2048 game board.

4.3.1 Detailed Description

Author: Saruggan Thiruchelvan (thirus6) Revised: April 11, 2021

Description: View Module (Abstract Object)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 11, 2021

Description: BoardUI Module (Abstract Data Type)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 11, 2021

Description: ComponentUI Module (Interface Module)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 11, 2021

Description: MessageUI Module (Abstract Data Type)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 11, 2021

Description: ScoreUI Module (Abstract Data Type)

Author: Saruggan Thiruchelvan (thirus6) Revised: April 11, 2021

Description: TileUI Module (Abstract Data Type)

Chapter 5

Class Documentation

5.1 src.controller.BoardManager Class Reference

A library containing useful functions for manipulating BoardT objects.

Static Public Member Functions

- static BoardT merge (BoardT board, DirectionT direction)

  Return a new BoardT object such that for every pair of duplicate tiles, the value of the tile in "front" is doubled and the value of the tile "behind" is set to 0 (i.e empty).
- static int scoreFromMerge (BoardT board, DirectionT direction)

  Calculate the points accumulated from merging the given board in the given direction.
- static BoardT align (BoardT board, DirectionT direction)

  Return a new BoardT object such that the every tile in the board is aligned in the given direction.
- static BoardT addRandomTile (BoardT board)

  Return a copy of the given BoardT object with a random tile added to a random empty position of the board.

5.1.1 Detailed Description

A library containing useful functions for manipulating BoardT objects.

Note: All methods do not change the state of the BoardT object passed in. Instead, a new BoardT object is returned.

5.1.2 Member Function Documentation

5.1.2.1 addRandomTile()

```
static BoardT src.controller.BoardManager.addRandomTile (
            BoardT board )  [static]
```

Return a copy of the given BoardT object with a random tile added to a random empty position of the board.

Note: The probability of the value of the random tile is 90% chance of a 2 and 10% chance of a 4.

Parameters

board	The board to add a random tile to.

Returns

A new BoardT object with a random tile added to a random position of the given board.

Exceptions

IllegalStateException	If the given board is full.

5.1.2.2 align()

```
static BoardT src.controller.BoardManager.align (
            BoardT board,
            DirectionT direction )  [static]
```

Return a new BoardT object such that the every tile in the board is aligned in the given direction.

Parameters

board	The board to be aligned in the given direction.
direction	The direction that the board is being aligned in.

Returns

A new BoardT object representing the board aligned in the given direction.

5.1.2.3 merge()

```
static BoardT src.controller.BoardManager.merge (
            BoardT board,
            DirectionT direction )  [static]
```

Return a new BoardT object such that for every pair of duplicate tiles, the value of the tile in "front" is doubled and the value of the tile "behind" is set to 0 (i.e empty).

Parameters

board	The board to be merged in the given direction.
direction	The direction that the board is being merged in.

Returns

A new BoardT object representing the board merged in the given direction.

5.1.2.4 scoreFromMerge()

```
static int src.controller.BoardManager.scoreFromMerge (
            BoardT board,
            DirectionT direction )  [static]
```

Calculate the points accumulated from merging the given board in the given direction.

For every pair of duplicate tiles, the amount of points accumulated is equal to sum of the value of both tiles (or double the value of the tile).

Parameters

board	The board to be merged in the given direction.
direction	The direction that the board is being merged in.

Returns

The number of points accumulated from merging the given board in the given direction.

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

- src/controller/BoardManager.java

5.2 src.model.BoardT Class Reference

An ADT that implements the board for the 2048 game using a matrix (2-dimensional array) of integer values to represent the state of the board.

Public Member Functions

- BoardT ()

  Initializes a BoardT object with all tiles set to 0 (i.e. empty).
- int getTile (int i, int j)

  Get the value stored at the tile at positon i, j of the board.
- void setTile (int i, int j, int value)

  Set the value to be stored at the tile at position i, j of the board.
- boolean isTileEmpty (int i, int j)

  Check if the value stored at the tile at position i, j of the board is 0 (i.e. empty).
- boolean isFull ()

  Check if the board is full.
- boolean equals (BoardT other)

  Check if board is equal to another given board.
- BoardT copy ()

  Make and return a copy of the current BoardT object.

Public Attributes

- final int SIZE = 4

    The width/height of the board.

5.2.1 Detailed Description

An ADT that implements the board for the 2048 game using a matrix (2-dimensional array) of integer values to represent the state of the board.

Assume the top-left position of the board is (i, j) = (0, 0) and assume the bottom-right position of the board is (i, j) = (SIZE-1, SIZE-1).

5.2.2 Member Function Documentation

5.2.2.1 copy()

```
BoardT src.model.BoardT.copy ( )
```

Make and return a copy of the current BoardT object.

Returns

A BoardT object that is equal to the current BoardT object

5.2.2.2 equals()

```
boolean src.model.BoardT.equals (
              BoardT other )
```

Check if board is equal to another given board.

Parameters

other	The board to compare equivalence with.

Returns

True if every tile in the board has the same value as very tile in other board.

5.2.2.3 getTile()

```
int src.model.BoardT.getTile (
            int i,
            int j )
```

Get the value stored at the tile at positon i, j of the board.

Parameters

i	The horizontal index of the tile.
j	The vertical index of the tile.

Returns

The value stored in the tile at position i, j of the board.

Exceptions

IndexOutOfBoundsException	If the value of i or j is not within the bounds of [0, SIZE).

5.2.2.4 isFull()

```
boolean src.model.BoardT.isFull ( )
```

Check if the board is full.

Returns

True if none of the tiles in the board are empty.

5.2.2.5 isTileEmpty()

```
boolean src.model.BoardT.isTileEmpty (
            int i,
            int j )
```

Check if the value stored at the tile at position i, j of the board is 0 (i.e. empty).

Parameters

i	The horizontal index of the tile.
j	The vertical index of the tile.

Returns

True if the value of the tile is 0.

Exceptions

IndexOutOfBoundsException	If the value of i or j is not within the bounds of [0, SIZE).

5.2.2.6 setTile()

```
void src.model.BoardT.setTile (
            int i,
            int j,
            int value )
```

Set the value to be stored at the tile at position i, j of the board.

Parameters

i	The horizontal index of the tile.
j	The vertical index of the tile.
value	The value to be stored in the tile at positition i, j of the board.

Exceptions

IndexOutOfBoundsException	If the value of i or j is not within the bounds of [0, SIZE).

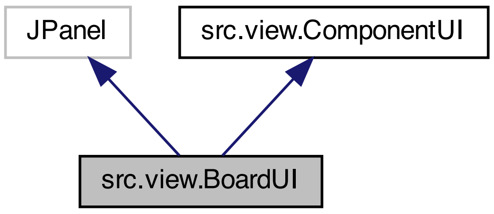
The documentation for this class was generated from the following file:

- src/model/BoardT.java

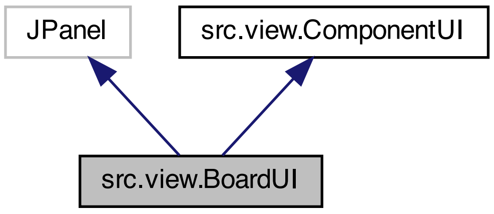
5.3 src.view.BoardUI Class Reference

An abstract object to display and update the view of the current state of the 2048 game board from the Controller module.

Inheritance diagram for src.view.BoardUI:



Collaboration diagram for src.view.BoardUI:



Public Member Functions

- BoardUI ()

  Initialize and setup the look of the JPanel as well as access the current BoardT instance from the Controller module to initialize the matrix (2-dimensional array) of TileUI components.
- void update ()

  Access the current BoardT instance from the Controller module and set/update each TileUI component.

5.3.1 Detailed Description

An abstract object to display and update the view of the current state of the 2048 game board from the Controller module.

This module extends the Java Swing JPanel class and implements the ComponentUI interface. The BoardT instance is accessed directly from the Controller module and displayed in a grid layout using TileUI components.

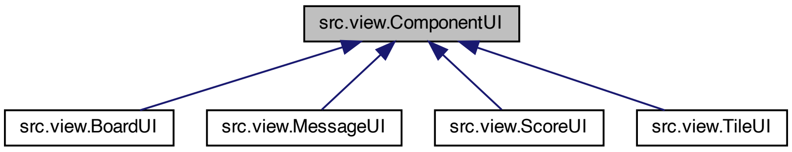
The documentation for this class was generated from the following file:

- src/view/BoardUI.java

5.4 src.view.ComponentUI Interface Reference

An interface for UI components.

Inheritance diagram for src.view.ComponentUI:



Public Member Functions

- void update ()

  The update method contains logic and functionality for updating the state of the UI component as well as updating the look of the UI component to reflect the changes.

5.4.1 Detailed Description

An interface for UI components.

UI components are modules that extend a Java Swing JComponent class that need to be updated regularly and/or have values that change with each move in the 2048 game. UI component modules have the suffix "UI" in their names.

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

- src/view/ComponentUI.java

5.5 src.controller.Controller Class Reference

An abstract object that represents a controller used to make moves and check the state of the 2048 game.

Static Public Member Functions

- static void newGame ()

  Reset the board, add two random tiles and reset the score.
- static void resumeGame (BoardT resumeBoard, int initialScore, int initialHighScore)

  Resume a previous/existing game given a board, score and high score.
- static void move (DirectionT direction)

  Make a move in the game in the given direction.
- static boolean canMove (DirectionT direction)

  Check if a move in the given direction changes the board.
- static boolean isGameOver ()

  Check if the game is over.
- static int nextTileToGet ()

  Get the next highest tile to be achieved in the game.
- static BoardT getBoard ()

  Return a copy of the current board.

5.5.1 Detailed Description

An abstract object that represents a controller used to make moves and check the state of the 2048 game.

The state of the board is stored in a BoardT object and the score is maintained using the ScoreT module. Either newGame or resumeGame is called before using the controller.

5.5.2 Member Function Documentation

5.5.2.1 canMove()

```
static boolean src.controller.Controller.canMove (
            DirectionT direction )  [static]
```

Check if a move in the given direction changes the board.

Parameters

direction	The direction to be moved in.

Returns

True if the board resulting from making a move in the given direction is not equal to the board prior to the move.

See also

move

5.5.2.2 getBoard()

```
static BoardT src.controller.Controller.getBoard ( )  [static]
```

Return a copy of the current board.

Returns

A BoardT object equal to the current board.

5.5.2.3 isGameOver()

```
static boolean src.controller.Controller.isGameOver ( )  [static]
```

Check if the game is over.

Returns

True if one cannot move in all four directions.

5.5.2.4 move()

```
static void src.controller.Controller.move (
             DirectionT direction )  [static]
```

Make a move in the game in the given direction.

Assuming canMove is true, the board is aligned, merged and aligned in the given direction and the score is updated accordingly (using the ScoreT module).

Parameters

direction	The direction to be moved in.

Exceptions

IllegalStateException	If the game is over.

See also

canMove
isGameOver

5.5.2.5 nextTileToGet()

```
static int src.controller.Controller.nextTileToGet ( )  [static]
```

Get the next highest tile to be achieved in the game.

At the start of a new game, the next tile to be achieved is 2048. After the 2048 tile is achieved then the next tile is 4096, then 8192, etc.

Returns

The value of the next highest tile to be achieved in the game.

5.5.2.6 resumeGame()

```
static void src.controller.Controller.resumeGame (
            BoardT resumeBoard,
            int initialScore,
            int initialHighScore )  [static]
```

Resume a previous/existing game given a board, score and high score.

This is to be used to restore the state of the game and continue to play.

Parameters

resumeBoard	The board to resume/continue the game from.
initialScore	The current score to resume/continue the game from (initialized using the ScoreT module).
initialHighScore	The high score to resume/continue the game from (initialized using the ScoreT module).

Exceptions

IllegalArgumentException	If the initial score or initial high score is less than 0 or if the initial high score is less than the initial score.

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

- src/controller/Controller.java

5.6 src.Demo Class Reference

A library containing a single method for assembling the model, view and controller modules and launching the playable 2048 application. Please comment/ uncomment one of the two sections in the main method to either start a new game or restore a previous game.

Static Public Member Functions

- static void main (String[ ] args)

  Initialize the model, view and controller objects, and launch the application.

5.6.1 Detailed Description

A library containing a single method for assembling the model, view and controller modules and launching the playable 2048 application. Please comment/ uncomment one of the two sections in the main method to either start a new game or restore a previous game.

5.6.2 Member Function Documentation

5.6.2.1 main()

```
static void src.Demo.main (
            String[] args )  [static]
```

Initialize the model, view and controller objects, and launch the application.

Choose to either initialize a new game of 2048 or resume/continue a previous game based on a user-defined configuration of the board (must explicitly comment/uncomment selection).

Parameters

args	

1) Initialize and display a new game of 2048.

2) Resume/restore a previous game of 2048.

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

- src/Demo.java

5.7 src.model.DirectionT Enum Reference

An enumerated type that represents the direction of a given move in the game (i.e. the arrow key directions).

Public Attributes

- **LEFT**
- **RIGHT**
- **UP**
- **DOWN**

5.7.1 Detailed Description

An enumerated type that represents the direction of a given move in the game (i.e. the arrow key directions).

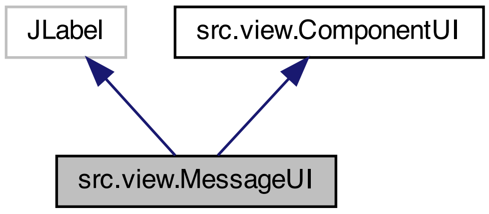
The documentation for this enum was generated from the following file:

- src/model/DirectionT.java

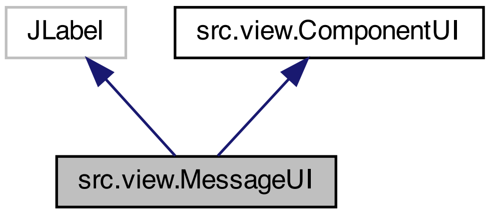
5.8 src.view.MessageUI Class Reference

An abstract object to display and update the view of the current state the 2048 game from the Controller module with respect to whether the player has achieved the 2048 tile or if the game is over.

Inheritance diagram for src.view.MessageUI:



Collaboration diagram for src.view.MessageUI:



Public Member Functions

- MessageUI ()

    Initialize and setup the look of the JLabel as well as access the current state of the game from the Controller module to set the text accordingly.
- void update ()

    Access the current state of the game from the Controller module and update the text accordingly.

5.8.1 Detailed Description

An abstract object to display and update the view of the current state the 2048 game from the Controller module with respect to whether the player has achieved the 2048 tile or if the game is over.

This module extends Java Swing JLabel class and implements the ComponentUI interface. The state of the game is accessed directly from the Controller module and displays it alongside basic instructions for the game.

5.8.2 Member Function Documentation

5.8.2.1 update()

```
void src.view.MessageUI.update ( )
```

Access the current state of the game from the Controller module and update the text accordingly.

The message displayed varies according to the state of the game. If the game is over, it prompts the player to start a new game or exit. Else, it prompts the player to use the arrow keys to a make a move to get the 2048 tile. If the 2048 tile has been achieved, it displays the next tile to get.

Implements src.view.ComponentUI.

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

- src/view/MessageUI.java

5.9 src.model.ScoreT Class Reference

An abstract object that maintains the score and high score of the game.

Static Public Member Functions

- static void initialize (int initialScore, int initialHighScore)
  Initialize the score and high score of the game.
- static void updateScore (int points)
  Add the given number of points to the current score and update the high score if necessary.
- static int getScore ()
  Get the value of the current score of the game.
- static int getHighScore ()
  Get the value of the current high score of the game.
- static void resetScore ()
  Reset the current score of the game back to 0.

5.9.1 Detailed Description

An abstract object that maintains the score and high score of the game.

5.9.2 Member Function Documentation

5.9.2.1 getHighScore()

```
static int src.model.ScoreT.getHighScore ( )  [static]
```

Get the value of the current high score of the game.

Returns

The current high score of the game.

5.9.2.2 getScore()

```
static int src.model.ScoreT.getScore ( )  [static]
```

Get the value of the current score of the game.

Returns

The current score of the game.

5.9.2.3 initialize()

```
static void src.model.ScoreT.initialize (
             int initialScore,
             int initialHighScore )  [static]
```

Initialize the score and high score of the game.

By default, the score and high score are set to 0 but this is used for purposes of resuming a previous game.

Parameters

initalScore	The initial score of the game.
initialHighScore	The initial high score of the game.

Exceptions

IllegalArgumentException	If the initial score or initial high score is less than 0 or if the initial high score is less than the initial score.

5.9.2.4 updateScore()

```
static void src.model.ScoreT.updateScore (
            int points )  [static]
```

Add the given number of points to the current score and update the high score if necessary.

Parameters

points	The number of points to be added to current score.

Exceptions

IllegalArgumentException	If number of points being added is less than 0.

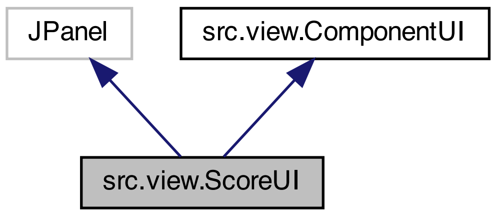
The documentation for this class was generated from the following file:

- src/model/ScoreT.java

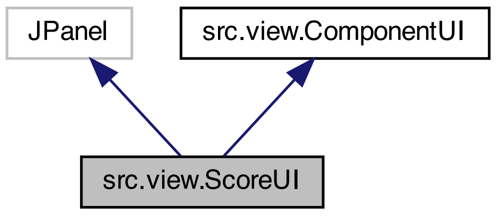
5.10 src.view.ScoreUI Class Reference

An abstract object to display and update the view of the current score and high score of the 2048 game from the ScoreT module.

Inheritance diagram for src.view.ScoreUI:



Collaboration diagram for src.view.ScoreUI:



Public Member Functions

- ScoreUI ()

  Initialize and setup the look of the JPanel as well as access the score/high score from the ScoreT module to display it using two JLabels.
- void update ()

  Access the current score/high score from the ScoreT module and update the text of the two JLabels.

5.10.1 Detailed Description

An abstract object to display and update the view of the current score and high score of the 2048 game from the ScoreT module.

This module extends Java Swing JPanel class and implements the ComponentUI interface. The score and high score are displayed using two JLabels and their values are accessed directly from the ScoreT module.

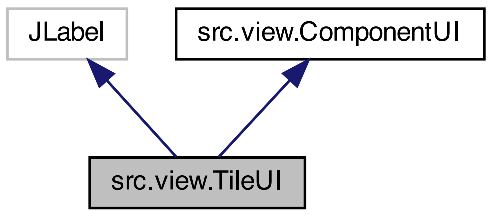
The documentation for this class was generated from the following file:

- src/view/ScoreUI.java

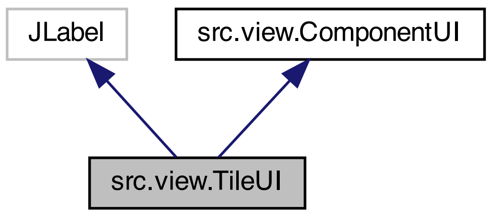
5.11 src.view.TileUI Class Reference

An abstract data type to display and update the view of an individual tile in the 2048 game board.

Inheritance diagram for src.view.TileUI:



Collaboration diagram for src.view.TileUI:



Public Member Functions

- TileUI (int initalValue)

  Initialize and setup the look of the JLabel as well as set its value and consequently its background color, foreground color and text accordingly.
- void setValue (int newValue)

  Set the current value of the tile. If the tile is "empty", its value is set to 0.
- void update ()

  Set the background color, foreground color, and text of the JLabel according to the current value of the tile.

5.11.1 Detailed Description

An abstract data type to display and update the view of an individual tile in the 2048 game board.

This module extends the Java Swing JLabel class and implements the ComponentUI interface. The value of the tile is set (by the BoardT module) and the background color, foreground color, and text is updated accordingly to resemble the look of the tiles in original 2048 game. If the tile is "empty", its value is set to 0. Assume that the value of the tile is always positive and never greater than 4 digits in length.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 TileUI()

```
src.view.TileUI.TileUI (
             int initalValue )
```

Initialize and setup the look of the JLabel as well as set its value and consequently its background color, foreground color and text accordingly.

Parameters

initalValue	The intial value of the tile. If the tile is "empty", the initial value is set to 0.

5.11.3 Member Function Documentation

5.11.3.1 setValue()

```
void src.view.TileUI.setValue (
             int newValue )
```

Set the current value of the tile. If the tile is "empty", its value is set to 0.

Parameters

newValue	The value to set the tile to.

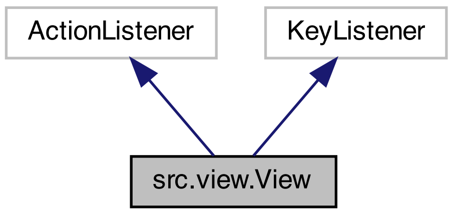
The documentation for this class was generated from the following file:

- src/view/TileUI.java

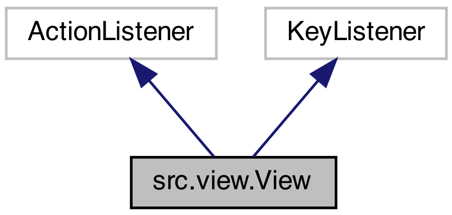
5.12 src.view.View Class Reference

An abstract object for displaying the the 2048 game using a graphical user interface (GUI).

Inheritance diagram for src.view.View:



Collaboration diagram for src.view.View:



Public Member Functions

- void **keyPressed** (KeyEvent e)
- void **keyTyped** (KeyEvent e)
- void **keyReleased** (KeyEvent e)
- void **actionPerformed** (ActionEvent e)

Static Public Member Functions

- static void display ()

  Initialize and display the GUI window.

5.12.1 Detailed Description

An abstract object for displaying the the 2048 game using a graphical user interface (GUI).

This module makes use of the Java Swing toolkit to generate GUI components and the Java AWT Event package to handle user interactions. The View module is represented by a host of UI component modules (that are updated regularly) such as BoardUI, ScoreUI,and MessageUI and some non-UI components such as a JFrame for the window and a JLabel for the title among others.

5.12.2 Member Function Documentation

5.12.2.1 display()

```
static void src.view.View.display ( )  [static]
```

Initialize and display the GUI window.

Assume this method is called after initializing the 2048 game using either newGame or resumeGame in the Controller module. A Java Swing JFrame acts a GUI window to display and update the BoardUI, ScoreUI, and MessageUI modules among other non-UI components.

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

- src/view/View.java

Index