Board.Board Class Reference

Public Member Functions

def __init__ (self)

def getBoard (self)

def printBoard (self)

Public Attributes

board

Detailed Description

```
Board class initializes the game board (Board uses the Tile class).

__init__: Input:
Function requires a parameter.
Output:
Function will initialize a 6x6 board with tiles.

getBoard: Input:
Function requires a parameter.
Output:
Function will return a board.

printBoard: Input:
```

Function requires a parameter.
Output:
Function will print a working board.

State variables:
self (for __init__): refers to newly created object
self (for getBoard): refers to instance whose method was called
self (for printBoard): refers to instance whose method was called.

Environment variables: None for this module.

Exceptions: None for this module.

Constructor & Destructor Documentation

def Board.Board.__init__ (*self*)

Constructor for Board.

Constructs a 6x6 board with tiles of random colours.

Member Function Documentation

def Board.Board.getBoard (*self*)

Getter method for Board.

def Board.Board.printBoard (*self*)

Method for printing the Board.

Member Data Documentation

Board.Board.board

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

- Board.py

Generated by 1.8.10

- Main Page
- Packages
- Classes
- Files
- Search

- Class List
- Class Index
- Class Members

- Logic
- Logic

Public Member Functions | List of all members

Logic.Logic Class Reference

Public Member Functions

def removeTile (board, row, col)

def checkColumn (columnNumber, board)

def moveDown (columnNumber, board, numEmpty)

def addTile (board)

def colourMatch (board, row, col, colour)

def adjacent (row1, col1, row2, col2)

def userInput (board)

Detailed Description

```
This class is responsible for
the logic behind the game.

removeTile: Input:
Function requires three parameters.
Output:
Function will remove tile in the given position.
```

checkColumn: Input:
Function requires two parameters.
Output:
Function will return number of empty spaces in a column.

moveDown: Input:
Function requires three parameters.
Output:
Function will move tiles down to empty positions in a column.

addTile: Input:
Function requires a parameter.
Output:
Function will add tile(s) to empty positions; this function uses checkColumn and moveDown methods.

colourMatch: Input:
Function requires four parameters.
Output:
Function will see if selected tiles are of the same colour or not (boolean).

adjacent: Input:
Function requires four parameters.
Output:
Function will see if selected tiles are adjacent to each other or not (boolean).

userInput: Input:
Function requires a parameter.
Output:
Function will make the board respond to user input.

State variables:
board: used by all, but one method in this module (adjacent does not need to use board)
row: represents a row on the board
col: represents a column on the board
columnNumber: refers to one column (out of 6) on the board
numEmpty: refers to the count of empty positions in a column
colour: refers to colour of a tile
row1 and row2: refer to any two rows on board

```
col1 and col2: refer to any two columns on the board

Environment variables:
row (in userInput method): based on user interaction with keyboard; user has to enter row number
col (in userInput method): user has to enter column number
contTurn (in userInput method): user can decide to continue with game or quit

Exceptions:
row (in userInput method): row number entered out of range
    Row number has to be between 1 and 6

col (in userInput method): column number entered out of range
    Column number has to be between 1 and 6

contTurn (in userInput method): character entered invalid
    Character entered has to be 'y' or 'n'
```

Member Function Documentation

def Logic.Logic.addTile (*board*)

```
Method for adding tile(s)
in empty position(s). This method
uses checkColumn and
moveDown methods as well.
```

def Logic.Logic.adjacent (*row1*,
col1,
row2,
col2
)

```
Method to check whether
```

```
the newly selected tile is
adjacent or not to the
currently selected one.
```

def Logic.Logic.checkColumn (*columnNumber*,
board
)

```
Method for checking empty
spaces in a column.
```

def Logic.Logic.colourMatch (*board*,
row,
col,
colour
)

```
Method to check if the color
of tiles is the same or not (when trying to match).
```

def Logic.Logic.moveDown (*columnNumber*,
board,
numEmpty
)

```
Method for moving tiles down
to empty positions in a given column.
```

def Logic.Logic.removeTile (*board*,
row,

col

)

```
Method for removing
a tile (in the case where tiles of same
color are matched).
```

def Logic.Logic.userInput (*board*)

```
Method for all possible
inputs that user can give
to board. Two important
actions include deleting
tiles by entering row(s) and
column(s), and continuing
with/ending the game.
```

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

- Logic.py

Generated by *1.8.10*

My Project

- Main Page
- Packages
- Classes
- Files
- Search

- Class List
- Class Index
- Class Members

- Main
- Main

Static Public Attributes | List of all members

Main.Main Class Reference

Static Public Attributes

logic = L.Logic

tuple board = L.B.Board()

Detailed Description

```
This class is the basis
for initializing the game
with a proper board and logic.

Input(s): None for this module.
Output(s): None for this module.

State variables: None for this module.
Environment variables: None for this module.

Exceptions: None for this module.
```

Member Data Documentation

tuple Main.Main.board = L.B.Board() static

Main.Main.logic = L.Logic static

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

- Main.py

Generated by 1.8.10

My Project

Tile.Tile Class Reference

Public Member Functions

def __init__ (self)

def getColor (self)

Public Attributes

color

Detailed Description

```
Class for representing and creating
tiles to populate the board with.

__init__: Input:
Function requires a parameter.
Output:
Function will initialize a tile of random color.

getColor: Input:
Function requires a parameter.
Output:
Function will return tile's color.

State variables:
self (for __init__): refers to newly created object
self (for getColor): refers to instance whose method was called
```

```
Environment variables: None for this module.

Exceptions: None for this module.
```

Constructor & Destructor Documentation

def Tile.Tile.__init__ (*self*)

```
Constructor method for
setting a random color to a tile.
```

Member Function Documentation

def Tile.Tile.getColor (*self*)

```
Getter method for tile color.
```

Member Data Documentation

Tile.Tile.color

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

- Tile.py

Generated by 1.8.10