ScrabbleGame

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 scrabbleGame.Frame Class Reference

Public Member Functions

- ArrayList< Tile > getTiles ()
- Frame (ArrayList< Tile > startingTiles)
- Frame (Pool p)
- Frame ()
- boolean containsTile (Tile letter)
- boolean containsTile (char letter)
- Tile getTile (char letter)
- Tile playTile (Tile letter)
- Tile discardTile (Tile letter)
- boolean isEmpty ()
- void refillFrame (Pool pool)
- String toString ()

2.1.1 Detailed Description

Frame Class

This class represents the Frame (Rack) in Scrabble. The frame contains up to 7 tiles. It has methods for removing and adding tiles. Team: JunkBot Members: Reuben Mulligan (18733589), Evan Spendlove (18492656), Cal Nolan(18355103)

Author

Evan Spendlove

Version

1.0.0

Since

07-02-2020

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Frame() [1/3]

```
scrabbleGame.Frame.Frame (
            ArrayList< Tile > startingTiles )
```

Constructor which takes an ArrayList of tiles as an argument for initialising tiles field.

Parameters

startingTiles	Pass this argument to initialise the frame with up to 7 tiles.

2.1.2.2 Frame() [2/3]

```
scrabbleGame.Frame.Frame (
            Pool p )
```

Constructor which takes an instance of Pool as an argument and initialises tiles field using refillFrame().

Parameters

p	Pass this argument to initialise the frame with tiles from the Pool.

2.1.2.3 Frame() [3/3]

```
scrabbleGame.Frame.Frame ( )
```

Empty constructor which initialises the tiles ArrayList to a new ArrayList with capacity for 7 tiles.

2.1.3 Member Function Documentation

2.1.3.1 containsTile() [1/2]

```
boolean scrabbleGame.Frame.containsTile (
            char letter )
```

Method used for checking if the rack contains a given tile.

Parameters

letter	Pass the letter that you want to check (Tile conversion done by method).

Returns

boolean Returns true if the rack contains the tile, else false.

2.1.3.2 containsTile() [2/2]

```
boolean scrabbleGame.Frame.containsTile (
            Tile letter )
```

Method used for checking if the rack contains a given tile.

Parameters

letter	Pass the tile that you want to check.

Returns

boolean Returns true if the rack contains the tile, else false.

2.1.3.3 discardTile()

```
Tile scrabbleGame.Frame.discardTile (
            Tile letter )
```

Method used for discarding a tile that you do not want.

Parameters

letter	Pass the tile that you do not want to keep.

Returns

Tile Returns the tile that you are discarding.

2.1.3.4 getTile()

```
Tile scrabbleGame.Frame.getTile (
            char letter )
```

Method used for getting a tile from the rack

Parameters

letter	Pass the letter that you want to get.

Returns

Tile Returns the Tile for the given letter on the rack.

2.1.3.5 getTiles()

```
ArrayList<Tile> scrabbleGame.Frame.getTiles ( )
```

Getter for tiles ArrayList.

Returns

ArrayList Returns the ArrayList of tiles stored in the Frame.

2.1.3.6 isEmpty()

```
boolean scrabbleGame.Frame.isEmpty ( )
```

Method used for checking if the rack is currently empty.

Returns

boolean Returns true if the rack is empty, else false.

2.1.3.7 playTile()

```
Tile scrabbleGame.Frame.playTile (
            Tile letter )
```

Method used for playing a tile on the board.

Parameters

letter	Pass the tile that you want to play.

Returns

Tile Returns the tile after removing it from the rack.

2.1.3.8 refillFrame()

```
void scrabbleGame.Frame.refillFrame (
            Pool pool )
```

Method used for refilling the frame to the full 7 tiles from a pool.

Parameters

pool	Pass the pool from which the letters will be drawn.

2.1.3.9 toString()

```
String scrabbleGame.Frame.toString ( )
```

Method overriding toString() from Object to allow custom String for printing

Returns

String Returns a custom String representation of this Class

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

- JunkBot/JunkBot/src/main/java/scrabbleGame/Frame.java

2.2 scrabbleGame.MainTest Class Reference

Static Public Member Functions

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

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

- JunkBot/JunkBot/src/main/java/scrabbleGame/MainTest.java

2.3 scrabbleGame.Player Class Reference

Public Member Functions

- Player (String username, int score, Frame frame)
- Player (String username, int score)
- Player (String username, Frame frame)
- Player (String username)
- String getUsername ()
- int getScore ()
- Frame getFrame ()
- void setFrame (Frame frame)
- void setUsername (String username)
- void setScore (int score)
- void resetPlayer ()
- String toString ()
- String dumpPlayerInfo ()

2.3.1 Detailed Description

Player Class

This class represents the Player in Scrabble. The player has a username, a score and their frame (rack). The class has the relevant getters and setters, and a reset() method. Team: JunkBot Members: Reuben Mulligan (18733589), Evan Spendlove (18492656), Cal Nolan(18355103)

Author

Reuben Mulligan

Version

1.0.0

Since

07-02-2020

2.3.2 Constructor & Destructor Documentation

2.3.2.1 Player() [1/4]

```
scrabbleGame.Player.Player (
            String username,
            int score,
            Frame frame )
```

This is the full constructor that initialises the username, score and frame.

Parameters

username	Pass the username of the Player as a String.
score	Pass the current score of the Player as an int.
frame	Pass the frame of the Player.

2.3.2.2 Player() [2/4]

```
scrabbleGame.Player.Player (
            String username,
            int score )
```

This is a partial constructor that initialises the username, score of the Player and the frame to a new Frame object.

Parameters

username	Pass the username of the Player as a String.
score	Pass the current score of the Player as an int.

2.3.2.3 Player() [3/4]

```
scrabbleGame.Player.Player (
            String username,
            Frame frame )
```

This is a partial constructor that initialises the username and frame of the Player and the score to 0.

Parameters

username	Pass the username of the Player as a String.
frame	Pass the frame of the Player.

2.3.2.4 Player() [4/4]

```
scrabbleGame.Player.Player (
            String username )
```

This is a partial constructor that initialises the username of the Player, the score to 0 and the frame to a new Frame object.

Parameters

username	Pass the username of the Player as a String.

2.3.3 Member Function Documentation

2.3.3.1 dumpPlayerInfo()

```
String scrabbleGame.Player.dumpPlayerInfo ( )
```

Method for dumping all of the information about the Player object as a String.

Returns

Returns a string containing the username, score and frame of the Player.

2.3.3.2 getFrame()

```
Frame scrabbleGame.Player.getFrame ( )
```

Getter for private frame field.

Returns

Frame Returns the Frame object from the Player instance.

2.3.3.3 getScore()

```
int scrabbleGame.Player.getScore ( )
```

Getter for private score field.

Returns

int Returns the score of the Player in int format.

2.3.3.4 getUsername()

```
String scrabbleGame.Player.getUsername ( )
```

Getter for private username field.

Returns

String Returns the username of the Player in String format.

2.3.3.5 resetPlayer()

```
void scrabbleGame.Player.resetPlayer ( )
```

Method to reset the entire Player object (username, score, frame).

2.3.3.6 setFrame()

```
void scrabbleGame.Player.setFrame (
            Frame frame )
```

Setter for the private frame field.

Parameters

frame	Pass the frame that you want to set.

2.3.3.7 setScore()

```
void scrabbleGame.Player.setScore (
            int score )
```

Setter for the private score field with error checking.

Parameters

score	Pass the score (in int format) that you want to set.

2.3.3.8 setUsername()

```
void scrabbleGame.Player.setUsername (
            String username )
```

Setter for the private username field with error checking.

Parameters

username	Pass the username (in String format) that you want to set.

2.3.3.9 toString()

```
String scrabbleGame.Player.toString ( )
```

This method overrides the default String method of the Object class.

Returns

Returns a String containing the username and score in a formatted manner.

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

- JunkBot/JunkBot/src/main/java/scrabbleGame/Player.java

2.4 scrabbleGame.Pool Class Reference

Public Member Functions

- void addTile (Tile t)
- Pool ()
- int size ()
- boolean isEmpty ()
- void reset ()
- Tile draw ()
- int getValue (char c)
- String toString ()

2.4.1 Detailed Description

Pool Class

This class represents the bag of tiles (called the pool) in Scrabble. The pool contains 100 tiles, each a letter in the chosen alphabet. For our implementation, we are using the 26-letter english alphabet, with two additional blank tiles. The pool can be reset and you can draw tiles from the pool. Team: JunkBot Members: Reuben Mulligan (18733589), Evan Spendlove (18492656), Cal Nolan(18355103)

Author

Cal Nolan

Version

1.0

Since

07-02-2020

2.4.2 Constructor & Destructor Documentation

2.4.2.1 Pool()

```
scrabbleGame.Pool.Pool ( )
```

This is the constructor for the Pool class. The constructor initialises the pool field and calls the resetPool() method.

2.4.3 Member Function Documentation

2.4.3.1 addTile()

```
void scrabbleGame.Pool.addTile (
            Tile t )
```

Public method for adding a tile to the pool.

Parameters

t	Takes a tile, t, as input and adds it to the pool.

2.4.3.2 draw()

```
Tile scrabbleGame.Pool.draw ( )
```

This method removes a randomly chosen tile from the pool and returns it.

Returns

Tile Returns the randomly chosen tile.

2.4.3.3 getValue()

```
int scrabbleGame.Pool.getValue (
            char c )
```

This method allows users to check the value of a tile by passing the character.

Parameters

c	The character for which you want to get the value (worth) of the tile.

Returns

int This returns the value of the Tile associated with this character.

2.4.3.4 isEmpty()

```
boolean scrabbleGame.Pool.isEmpty ( )
```

This method is used to test if the pool contains any tiles.

Returns

boolean This returns true if the pool is empty, else it returns false.

2.4.3.5 reset()

```
void scrabbleGame.Pool.reset ( )
```

This method is used to reset the pool to its original state, containing all tiles.

2.4.3.6 size()

```
int scrabbleGame.Pool.size ( )
```

This method is used to get the current size of the pool (i.e. the remaining tiles).

Returns

int This returns the size of the pool field.

2.4.3.7 toString()

```
String scrabbleGame.Pool.toString ( )
```

This method overrides the default String method of the Object class.

Returns

String Returns the String format of the pool ArrayList;

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

- JunkBot/JunkBot/src/main/java/scrabbleGame/Pool.java

2.5 scrabbleGame.Tile Enum Reference

Public Member Functions

- Tile (char c, int value)
- char character ()
- int value ()
- String toString ()

Static Public Member Functions

- static Tile getInstance (char c)
- static int getValue (char c)
- static int getValue (String letter)

Public Attributes

- **BLANK** =('#', 0)
- **A** =('A', 1)
- **B** =('B', 3)
- **C** =('C', 3)
- **D** =('D', 2)
- **E** =('E',1)
- **F** =('F', 4)
- **G** =('G', 2)
- **H** =('H',4)
- **I** =('I', 1)
- **J** =('J', 8)
- **K** =('K', 5)
- **L** =('L', 1)
- **M** =('M', 3)
- **N** =('N', 1)
- **O** =('O', 1)
- **P** =('P', 3)
- **Q** =('Q', 10)
- **R** =('R', 1)
- **S** =('S', 1)
- **T** =('T', 1)
- **U** =('U', 1)
- **V** =('V', 4)
- **W** =('W', 4)
- **X** =('X', 8)
- **Y** =('Y', 4)
- **Z** =('Z', 10)

2.5.1 Detailed Description

Tile Class

This represents a tile in Scrabble, which has an associated character and value.

Team: JunkBot

- Members: Reuben Mulligan (18733589), Evan Spendlove (18492656), Cal Nolan(18355103)

  Author

  Evan Spendlove

  Version

  1.0.0

  Since

  07-02-2020

2.5.2 Constructor & Destructor Documentation

2.5.2.1 Tile()

```
scrabbleGame.Tile.Tile (
            char c,
            int value )
```

Default constructor for an enum class.

Parameters

c	takes a character as input for the constructor
value	takes a value as input for the constructor

2.5.3 Member Function Documentation

2.5.3.1 character()

```
char scrabbleGame.Tile.character ( )
```

This accessor allows access to the character field of the current Tile instance.

Returns

char this returns the character field of this current Tile instance

2.5.3.2 getInstance()

```
static Tile scrabbleGame.Tile.getInstance (
            char c )  [static]
```

This method returns and instance of the Tile class for the character passed

Parameters

c	the letter for which you want an instance of Tile

Returns

Tile This returns an instance of the Tile class for the character passed.

2.5.3.3 getValue() [1/2]

```
static int scrabbleGame.Tile.getValue (
            char c )  [static]
```

This method returns the value associated with the character passed as per the enum.

Parameters

c	pass a character for which you want the associated value.

Returns

int Returns the value associated with the character passed.

2.5.3.4 getValue() [2/2]

```
static int scrabbleGame.Tile.getValue (
            String letter )  [static]
```

This method returns the value associated with the String passed as per the enum.

Parameters

letter	pass a String for which you want the associated value.

Returns

int Returns the value associated with the String passed.

2.5.3.5 toString()

```
String scrabbleGame.Tile.toString ( )
```

This method overrides the toString() method of object for a custom String return value.

Returns

String Returns only the character of the current Tile instance

2.5.3.6 value()

```
int scrabbleGame.Tile.value ( )
```

This accessor allows access to the value field of the current Tile instance.

Returns

int this returns the value field of this current Tile instance

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

- JunkBot/JunkBot/src/main/java/scrabbleGame/Tile.java

Index