Boards Class Reference

Public Member Functions

void	**displayBoard** () const
bool	**isValid** (char, int)
bool	**isHit** (char, int)
bool	**isPlaceable** (char, int)
bool	**isPlaceableRange** (char, char, int, int)
int	**charConvert** (char)
char	**intConvert** (int)
void	**displayHidden** () const
void	**shipCheck** (int row1, int row2, char col1, char col2, int size)
void	**placeShip** (int row, char col)
void	**getNumberOfShips** ()
bool	**isGameOver** ()
void	**startover** ()
void	**checkShot** (char X, int Y)
void	**FireHit** (char column, int row)
void	**FireMiss** (char column, int row)
void	**replace** (char a, int b)

Private Attributes

char	**myBoard** [8][8] Create a 2D array with 8*8 size.
int	**rows** Allow user to put the coordinates of placing ship.
int	**cols**
int	**numberOfShips** Allow user to choose the ship number.

Member Function Documentation

◆ charConvert()

int Boards::charConvert (char temp)

Precondition
A letter is read in for our guess.

Postcondition
Returns the numerical value of the letter.

◆ checkShot()

```
void Boards::checkShot ( char X,
                         int  Y
                       )
```

Precondition
Boards are filled.

Postcondition
Shot is registered as hit or miss.

◆ displayBoard()

void Boards::displayBoard ( ) const

Precondition
Board is kept safe in myBoard.

Postcondition
Board is displayed on screen.

◆ displayHidden()

```
void Boards::displayHidden ( ) const
```

Precondition
Board is filled with ships and water.

Postcondition
Board is displayed to screen, but ships are hidden by water.

◆ FireHit()

```
void Boards::FireHit ( char  column,
                       int   row
                     )
```

Precondition
Needs valid coordinate for fire

Postcondition
There is ship in that given location then shows hit

◆ FireMiss()

```
void Boards::FireMiss ( char  column,
                        int   row
                      )
```

Precondition
Needs valid coordinate for fire

Postcondition
There isn't any ship in that given location then shows miss

◆ getNumberOfShips()

void Boards::getNumberOfShips ( )

Precondition
none.

Postcondition
Number of ships desired for the game is received from user.

◆ intConvert()

char Boards::intConvert (int temp)

Precondition
A number is used for checking.

Postcondition
Number is changed back into it's letter.

◆ isGameOver()

bool Boards::isGameOver ( )

Precondition
none.

Postcondition
Game is over when none of ships are left.

◆ isHit()

```
bool Boards::isHit ( char  column,
                     int   row
                   )
```

Precondition
Board is filled with water and ships.

Postcondition
If guess is a ship, then return true.

Note
Turns the ship char into a hit char.

◆ isPlaceable()

```
bool Boards::isPlaceable ( char  col,
                           int   row
                         )
```

Precondition
Board has ships and water.

Postcondition
Returns true if it's water, false if it's a ship.

◆ isPlaceableRange()

```
bool Boards::isPlaceableRange ( char  col1,
                                char  col2,
                                int   row1,
                                int   row2
                              )
```

Precondition
Board has ships and water.

Postcondition
Returns true if all spaces are water in range.

Note
Calls isPlaceable over a range of spaces.

◆ isValid()

```
bool Boards::isValid ( char  column,
                       int   row
                     )
```

Precondition
None.

Postcondition
If the spot is valid to be attacked, true is returned

◆ placeShip()

```
void Boards::placeShip ( int   row,
                         char  col
                       )
```

Precondition
Board filled with water.

Postcondition
Ship piece placed at given row and col.

◆ replace()

```
void Boards::replace ( char  a,
                       int   b
                     )
```

Precondition
Needs valid coordinate for fire

Postcondition
replce the hitted ship with another char

◆ shipCheck()

```
void Boards::shipCheck ( int   row1,
                         int   row2,
                         char  col1,
                         char  col2,
                         int   size
                       )
```

Precondition
Board is filled with water, and maybe ships.

Postcondition
Board checks for validity on placement and places if possible.

◆ startover()

```
void Boards::startover ( )
```

Precondition
none.

Postcondition
give user a choice to start over

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

- C:/Users/Qing Dong/Desktop/Battleship-master/**Boards.h**
- C:/Users/Qing Dong/Desktop/Battleship-master/Boards.cpp
