Players Class Reference

Public Member Functions

	Players () creates the player, which has access to their boards.
	~Players () Frees up any assests the player has from memory.
void	**getBoards** () const Grabs both boards to display. More...
void	**getOffensiveBoard** () const Displays the board from an offensive standpoint.
void	**getDefensiveBoard** () const Displays the board from a defensive standpoint.
void	**getColumn** () Gets which column the user wants the ship in.
void	**getRow** () Gets which row the user wants the ship in.
void	**setShips** (int)
void	**setID** (std::string)
std::string	**getID** () Returns the player name to the called function.
bool	**getHit** (char, int)
bool	**shipsSet** () If all ships are set, return true. More...
bool	**hasLost** () If the player has no more ships, return true.
void	**markMyHits** (char, int) Marks my board with hits.
void	**markMyMisses** (char, int) Marks my board with misses.
void	**markTheirHits** (char, int) Marks the opponent's board with hits.
void	**markTheirMisses** (char, int) Marks the opponent's board with misses.
int	**charConvert** (char)
void	**cleanBoard** ()

Private Attributes

Boards *	**myBoard** Creating an object to access to board class.
std::string	**ID** String that save player's desire name.
int	**row** row number for hitting ships
char	**column** col char for hitting ships
bool	**allSet** bool that to determine whether players are ready to play

Member Function Documentation

◆ charConvert()

int Players::charConvert (char temp)

Precondition
Column character is given by user.

Postcondition
Character is converted into a numerical value.

◆ cleanBoard()

void Players::cleanBoard ( )

Precondition
Game is finished, and another is wanting to begin.

Postcondition
Board is cleaned off.

◆ getBoards()

void Players::getBoards ( ) const

Grabs both boards to display.

Precondition
None.

◆ getHit()

bool Players::getHit (char column,
int row
)

Precondition
Ships are placed on the board.

Postcondition
Returns true if a ship is at that location.

◆ setID()

void Players::setID (std::string name)

Precondition
Name desired for the player is given in exec.

Postcondition
Name is placed in storage for use by **Players**.

◆ setShips()

void Players::setShips (int number)

Precondition
Number of ships being placed known.

Postcondition
All ships are placed on the board.

◆ shipsSet()

bool Players::shipsSet ( )

If all ships are set, return true.

Precondition

None.

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

- C:/Users/Qing Dong/Desktop/EECS_448_Seg_Faults-master/**Players.h**
- C:/Users/Qing Dong/Desktop/EECS_448_Seg_Faults-master/Players.cpp