

Executive Class Reference

Public Member Functions

Executive ()

~Executive ()

void **displayLogo** ()

void **displayMenu** ()

void **run** ()

void **setPlayer1Name** ()

void **setPlayer2Name** ()

void **getP1Name** ()

void **getP2Name** ()

void **getNumberOfShips** ()

void **getColumn** ()

void **getRow** ()

void **clearScreen** ()

clears the screen.

void **pressToContinue** ()

Let's the user see if they hit or miss, then continues.

Private Attributes

int **row**

Get the row number from user, Get the choice using in interface.

int **choice**

char **column**

Get the col char from user.

Players * **player1**

Create an object to access to player class.

Players * **player2**

Create another object to access to player class for another player.

int **numberOfShips**

Get the number of ships from user.

Constructor & Destructor Documentation

◆ Executive()

Executive::Executive ()

Precondition

none

Postcondition

primary constructor of the **Executive** class

returns nothing

◆ ~Executive()

Executive::~~Executive ()

Precondition

none

Postcondition

destructor of the **Executive** class

returns nothing

Member Function Documentation

◆ displayLogo()

void Executive::displayLogo ()

Precondition

none

Postcondition

Displays the logo for the game

◆ displayMenu()

```
void Executive::displayMenu ( )
```

Precondition

none

Postcondition

Displays the menu for the game

◆ getColumn()

```
void Executive::getColumn ( )
```

Precondition

none

Postcondition

Repeatedly prompts the user for a column letter until a valid column letter is entered

◆ getNumberOfShips()

```
void Executive::getNumberOfShips ( )
```

Precondition

none

Postcondition

Prompts the user to enter the number of ships to be used during gameplay. Between 1 and 5 ships are allowed

◆ getP1Name()

```
void Executive::getP1Name ( )
```

Precondition

none

Postcondition

Returns the name of player 1 for display purposes

◆ getP2Name()

```
void Executive::getP2Name ( )
```

Precondition

none

Postcondition

Returns the name of player 2 for display purposes

◆ getRow()

```
void Executive::getRow ( )
```

Precondition

none

Postcondition

Repeatedly prompts the user for a row number until a valid row number is entered

◆ run()

```
void Executive::run ( )
```

Precondition

none

Postcondition

Handles the majority of functionality within the game. Gets player's name, calls the setShip method to begin setting the ships, and handles swapping turns for each player until the game is won

◆ setPlayer1Name()

```
void Executive::setPlayer1Name ( )
```

Precondition

none

Postcondition

Prompts the user for a name for player 1

◆ setPlayer2Name()

```
void Executive::setPlayer2Name ( )
```

Precondition

none

Postcondition

Prompts the user for a name for player 2

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

- C:/Users/Qing Dong/Desktop/EECS_448_Seg_Faults-master/[Executive.h](#)
- C:/Users/Qing Dong/Desktop/EECS_448_Seg_Faults-master/Executive.cpp