DJ Cmar

CIST 2361

7/1/2017

Final Project – Battleship

///////////////////////////////////////////////////////////////

// \_\_\_\_\_\_ \_ \_ \_ \_ \_ //

// | \_\_\_ \ | | | | | | | | (\_) //

// | |\_/ / \_\_ \_ | |\_ | |\_ | | \_\_\_ \_\_\_ | |\_\_ \_ \_ \_\_ //

// | \_\_\_ \ / \_` || \_\_|| \_\_|| | / \_ \/ \_\_|| '\_ \ | || '\_ \ //

// | |\_/ /| (\_| || |\_ | |\_ | || \_\_/\\_\_ \| | | || || |\_) | //

// \\_\_\_\_/ \\_\_,\_| \\_\_| \\_\_||\_| \\_\_\_||\_\_\_/|\_| |\_||\_|| .\_\_/ //

// CIST 2361 | | //

// A program written by: DJ Cmar |\_| //

///////////////////////////////////////////////////////////////

Battleship is a game where you and an opponent guess where each other have placed a set of five ships on a 10 by 10 grid. Each turn, a player would call out where they would like to “fire a missile” the other player says if the missile hits or misses the spot. When a ship gets hit in all possible spots, the player calls out which boat has been sunk. When each boat has been sunk, the opposing player wins.

This is the general layout I plan to use provided these Unicode keys are compatible with my program:

╔══════════════════════════════════════════════════════════╗

║ Computer launches missile at D1 and hits ║

╚══════════════════════════════════════════════════════════╝

╔═════════╦═══╗╔═══╦═══╦═══╦═══╦═══╦═══╦═══╦═══╦═══╦═══╦═══╗

║ T U R N ║1 3║║ ║ 1 ║ 2 ║ 3 ║ 4 ║ 5 ║ 6 ║ 7 ║ 8 ║ 9 ║ 0 ║

╠═════════╩═══╣╠═══╬═══╪═══╪═══╪═══╪═══╪═══╪═══╪═══╪═══╪═══╣

║plr1 miss I7 ║║ A ║ │ │ │ │ │ ◙ │ │ │ │ ║

║---Turn 13---║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║com1 hit D1 ║║ B ║ │ │ X │ X │ X │ ◙ │ │ O │ O │ ║

║plr1 miss I7 ║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║ ║║ C ║ O │ │ │ │ │ ◙ │ │ │ │ ║

║---Turn 12---║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║com1 miss H6 ║║ D ║ X │ │ │ │ │ ◙ │ │ │ │ ║

║plr1 hit B2 ║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║\*\*sink Boat\*\*║║ E ║ ◙ │ │ │ │ │ │ │ │ │ ║

║ ║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║---Turn 11---║║ F ║ │ ◙ │ ◙ │ X │ ◙ │ ◙ │ │ │ │ ║

║com1 hit B3 ║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║ \*\*sink\*\* ║║ G ║ │ │ │ │ │ │ │ │ O │ ║

║plr1 hit A8 ║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║ \*\*sink\*\* ║║ H ║ │ │ │ │ │ O │ │ │ │ ║

║ ║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║---Turn 10---║║ I ║ │ │ O │ │ O │ │ │ O │ │ ║

║com1 miss B9 ║╠═══╫───┼───┼───┼───┼───┼───┼───┼───┼───┼───╢

║plr1 hit A3 ║║ J ║ │ │ │ │ │ │ │ │ │ ║

╚═════════════╝╚═══╩═══╧═══╧═══╧═══╧═══╧═══╧═══╧═══╧═══╧═══╝

╔══════════════════════════════════════════════════════════╗

║ ║

║ ║

╚══════════════════════════════════════════════════════════╝

Otherwise, I will use this layout (which is practically the same, but not as visually stunning)

+=========================================================+

| |

+=========================================================+

+============++===========================================+

| T U R N || | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |

+============++===+===+===+===+===+===+===+===+===+===+===]

| || A | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || B | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || C | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || D | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || E | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || F | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || G | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || H | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || I | | | | | | | | | | |

| |[===+---+---+---+---+---+---+---+---+---+---]

| || J | | | | | | | | | | |

+============++===^===^===^===^===^===^===^===^===^===^===+

+=========================================================+

| |

| |

+=========================================================+

-My game board features a 30 line length and a 60 character width and will refresh itself anytime movement is made on it.

-The top display is mainly used for updating the player a live feed of current events

-The top and bottom display shows the player info used to help navigate or messages to let the player know if ships can be placed somewhere or if they are trying to target the same location twice.

-the left side box is used for keeping track of the most recent events.

MODULES/INSTANCES:

My program will feature many modules to re-use much of the code I plan to implement repeatedly. Here is a list of some of the main ones I plan to use as well as a short description of how they work.

**Main Modules:**

Init – initialization kicks off the start of the program. Variables will be initialized, game boards cleared…

Proc – Process will house all the modules that repeat themselves until a condition is met (quitting the game)

Eoj – End of job will wrap up anything that is needed after the program is ready to close (ensure files get saved)

**Mode Modules:**

menu – menu mode for navigating through the main menu. Calls vsPlayer and vsComputer modes

inputMode – most called module. Waits for key input from user. Actions are based on which key is pressed

vsComputer / vsPlayer – designed to set up a game vs a player or computer. calls setup modules and then turn modules.

setupPlayer1/ setupComputer / setupPlayer2 – has user informed on how the game works and also calls place modes.

PlaceMode / comPlaceShip – a mode designed for players to put all of their ships down on the map.

resetBoard – resets players game board arrays to a blank slate of all 0’s

whoGoesFirst – determines who goes first.

takeTurn – module that will get looped over and over again until someone achieves victory

checkForSink/checkForVictory – modules designed to see if a ship or all ships have been sunk

comAttack – randomly generates a hit or miss for the computer

**Display Modules:**

loadImage – loads an image from a file and uploads it to display array

display Image – clears the screen and displays the image from the display array

manipPixel – changes one spot on the display array to a different character

updateMap – changes the display array to show icons from a player’s 10 by 10 array

changeTopText – clears, changes, and centers the text at the top to whatever is inserted into the argument

clearTopText – clear top textbox

changeBottomText – clears, changes, and centers the text at the bottom to whatever two strings are inserted into the argument. The second string is default to being blank if only one argument is found.

clearBottomText – clears bottom textbox

addToInfoBox – adds a string argument to the left info box and then pushes all messages down by 1

clearInfoBox – clears the info box

highlightBoard – highlights a square on the game board

highlightPlacement- highlights two squares for ship placement