

David Haro

BattleShip

Version: 2.5

Project 1

CSC 17A

Prepared for: Dr. Mark Lehr

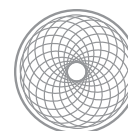
Prepared by: David Haro

October 26th, 2014

Attachments

Doxygen files

BattleShip V2_5 netbeans project folder



David Haro

Summary

Objective

Successfully plan and design a game in form of a C++ program that exemplifies the learning of this course. Methods and strategies that are desired are those that have been recently reviewed. This includes, but not limited to: Single and multidimensional arrays, dynamic memory allocation in 2D arrays, passing dynamic 2D arrays to functions, clearing memory when complete, and use of structs/

Goals

Develop and improve a previous version of Battleship/Salvo Game by using C++ knowledge. Fine tune methods that were previously covered during lecture, lab, and homework assignments.

The Game

The Game puts the player against the Computer in a game of Battleship. This version of the game is based off of the paper version once called Salvo. First the user must read the rules or directions is unfamiliar with game, then proceed with choosing coordinates where ships are to be deployed.

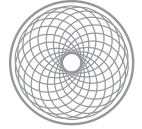
The Coordinates are put through series of validations in order to determine that they do not fall out of bounds or criss cross over other ships. Once the CPU selects its positions, the players are then asked to pick a strike point, and this is a mere hit or miss function.

The game ends when one of the players is out of ships on the board. The Program ends with displaying the winner.

Project Size

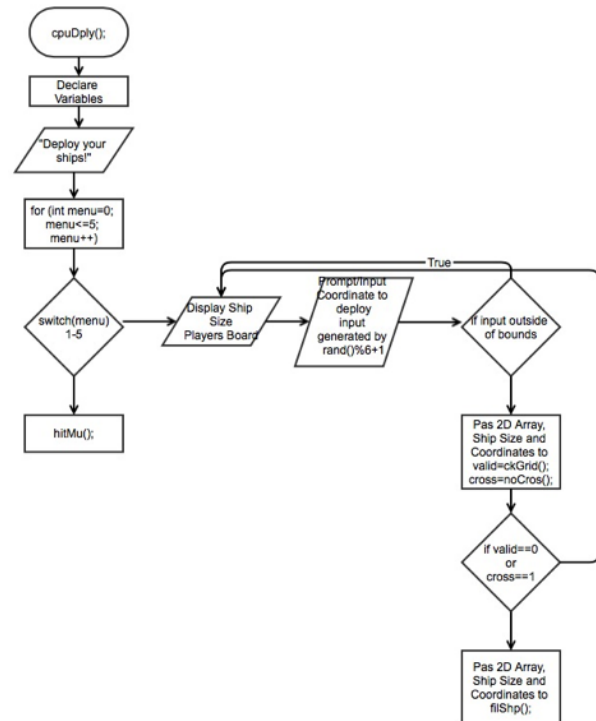
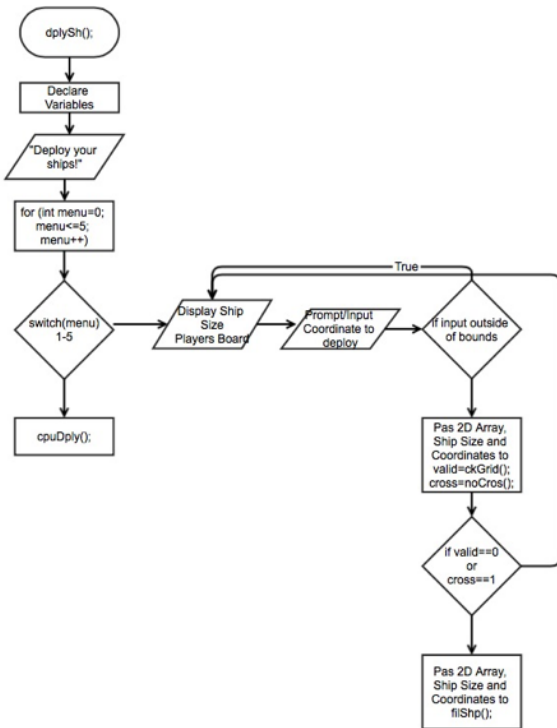
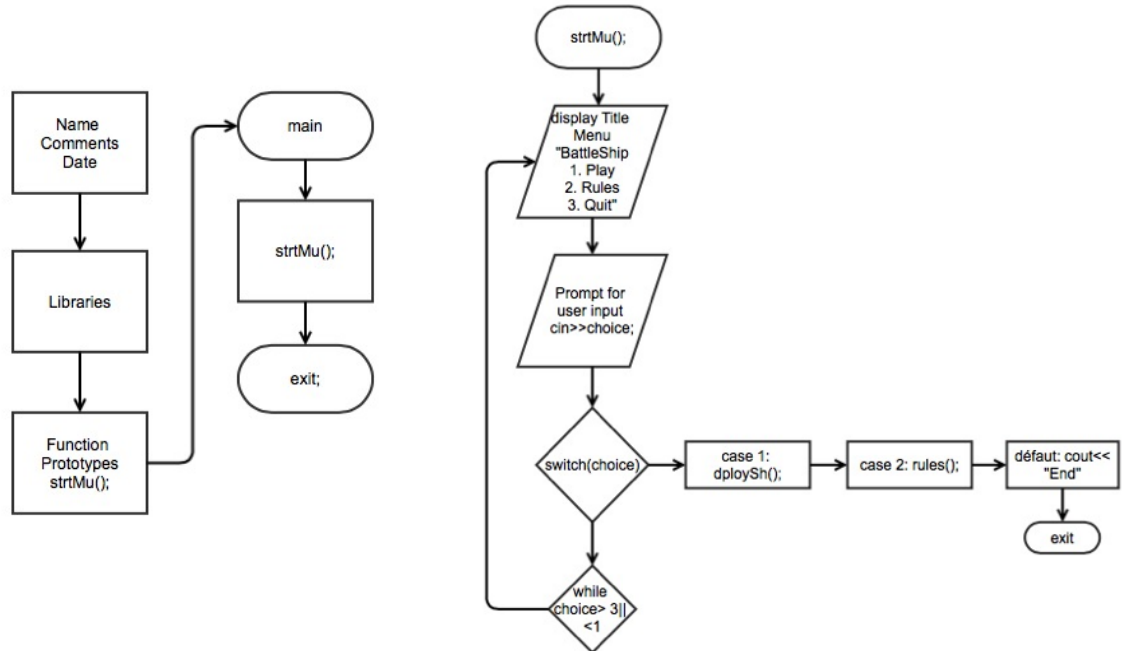
- Roughly 890 lines of code, this include about 40 lines of comments
- Number of 2D Arrays: 3
- Dynamically allocated arrays: 2
- Methods Used: Functions, returning Primitive data types, passing functions, single arrays, linear searching of 2D Arrays, structs, dynamic memory allocation etc.

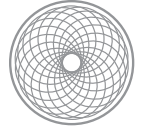
	A	B	C	D	E	F	G	H	I	L
1										
2										
3										
4			X							
5						X	X			
6		X						X		X
7				X						X
8	X	X						X		
9										
10										



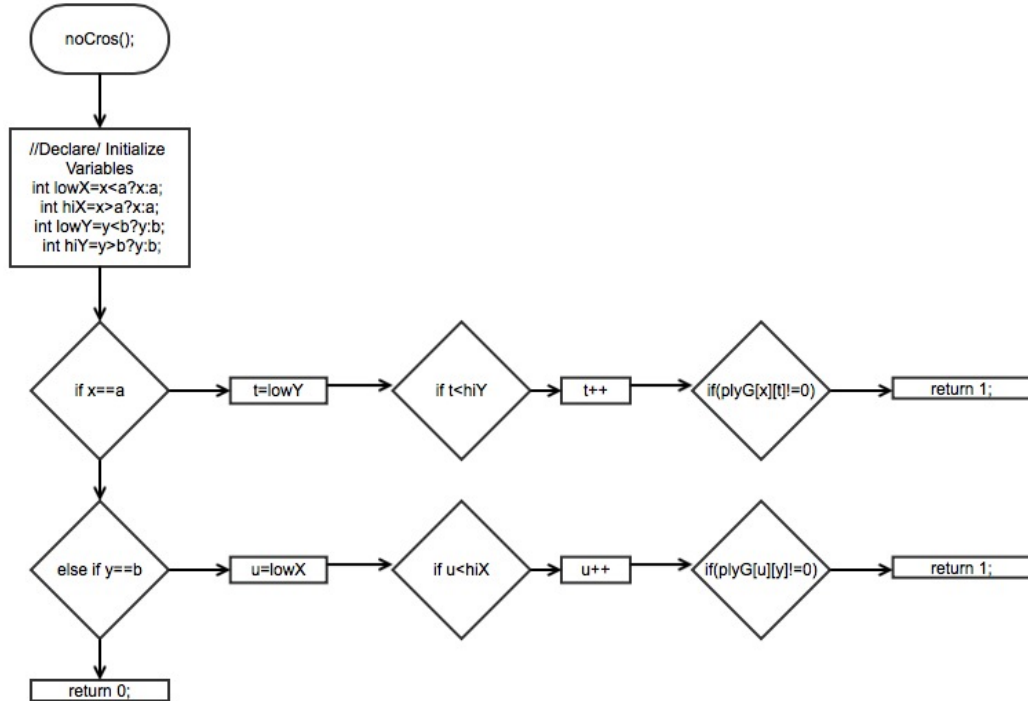
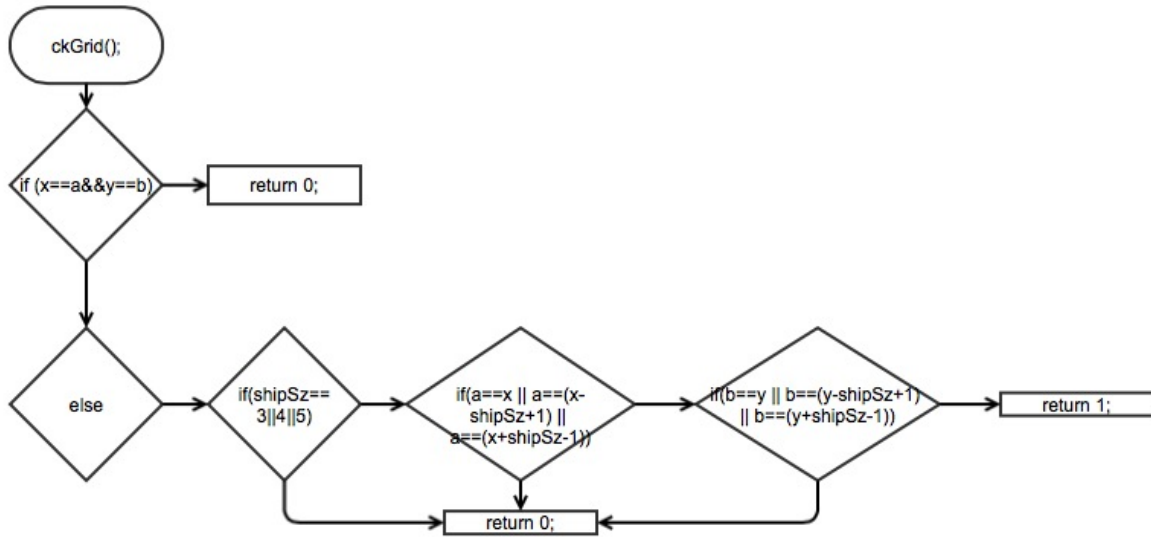
David Haro

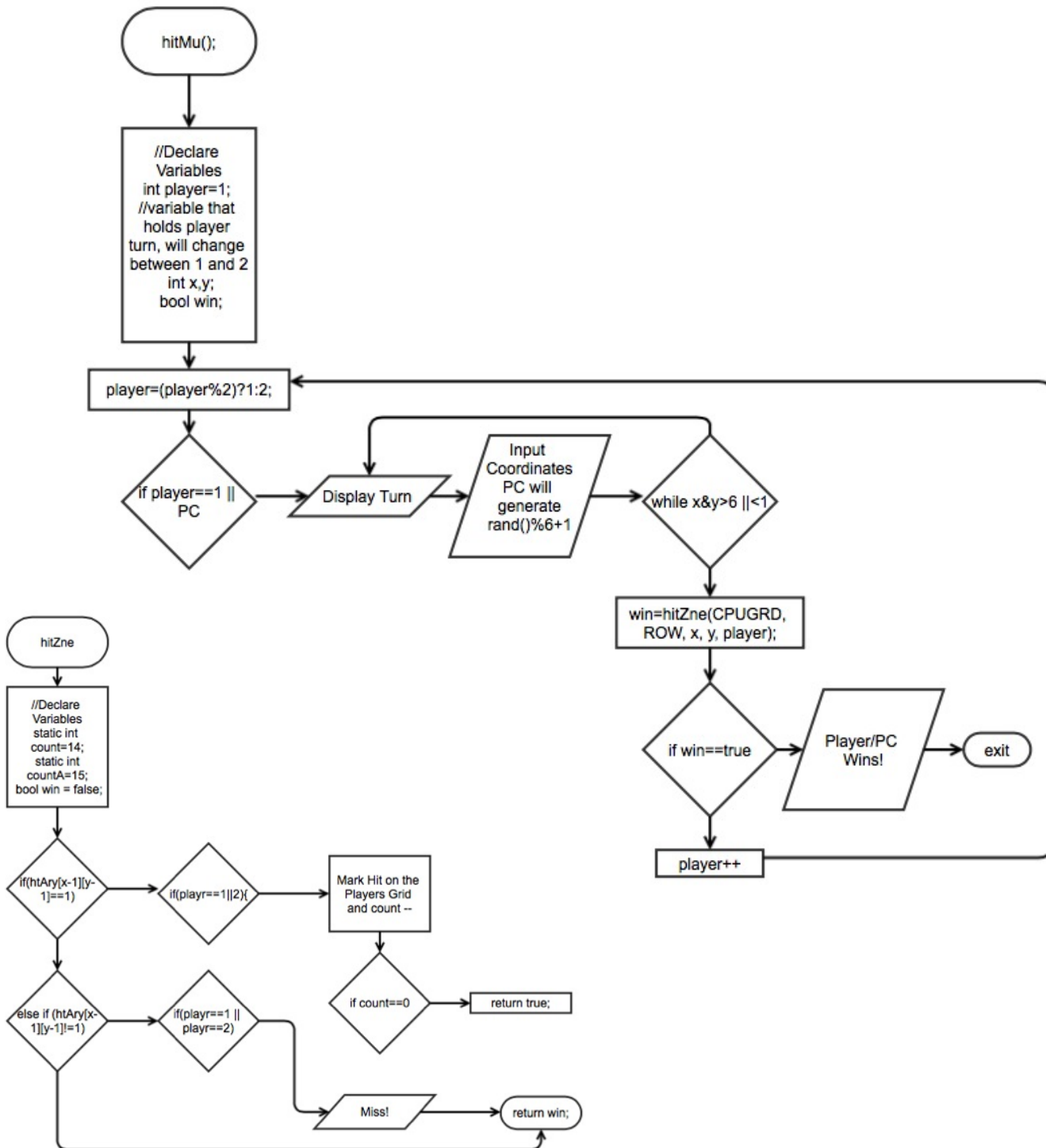
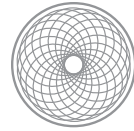
FlowCharts

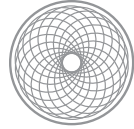




David Haro



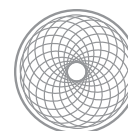




David Haro

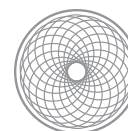
The Game: Broken Down

Type	Variable Name	Description
Library	iostream	allows to input and output data
	cstdlib	c standard library: allows to shorten code and references C language for code.
	iomanip	allows to format output
	ctime	needed for random number generator
constant int	ROW	needed to initialize number of ROWS in 2D array
	COL	needed to initialize number of COLS in 2D array
int	HITGRID[ROW][COL]	2D Array for Viewing Players Hit Markers
Functions	int main();	Initial function. Purpose: Call strtMu();
	void strtMu()	Displays Name of Game, provides menu for play, rules and quit. Allocates and initializes (2) 2D arrays and passes to function calls.
	void bsBoard(int **);	Receives dynamically allocated 2D array then displays Player's Grid.
	void cpuGrid(int **);	Receives dynamically allocated 2D array then displays CPU's Grid if player hits a ship part.
	void dploySh(int **, int **)	Function that Deploys ships
	int rules()	Function for Game Rules
	bool ckGrid(int **, int, int, int, int, int)	Checks to make sure coordinates are in range, returns a boolean value.

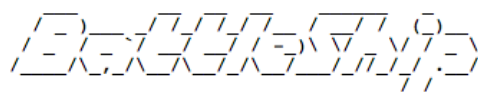


David Haro

Type	Variable Name	Description
	<code>void fillShp(iint **, int, int, int, int, int)</code>	Fills a series of elements for respective ships.
Functions (cont.)	<code>bool noCros(int **, int, int, int, int)</code>	Check criss crossing ships. Returns a boolean value.
	<code>void cpuDply(int **, int **)</code>	Generate CPU ship grid with 2D dyna-mem array
	<code>int hitMu(int **, int **)</code>	Toggle player turn/Hit Menu
	<code>bool hitZne(int **, int, int, int, int)</code>	Checks if coordinate is a valid strike zone. Returns a boolean value
	<code>void hitGrd()</code>	Blank Grid to mark Hit Spots for Player to view
	<code>void destroy(int **,int)</code>	Receives both dynamic 2D arrays and destroys them prior to ending game.
Structure	<code>struct Legend{ string lgnd; int hitMrk; int ship;};</code>	Created to hold the legend in the display of any grids containing player/ CPU hits to the left go the board.



David Haro



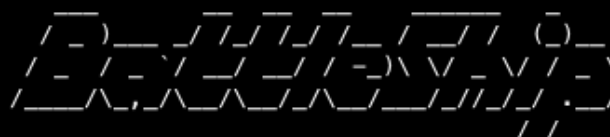
1. Play Game
2. See Rules
3. Quit

Make a Selection and Press 'Enter'



Sample Images

For Best results, make window full screen.



Programmed by David Haro

1* Play Game

2* See Rules

3* Quit

Make a Selection and Press 'Enter'

Rules for BattleShip

Each player selects an area to place their ships, whether in horizontal or vertical.

If a player strikes a spot in the grid that contains a part of the ship

, an '3' will be displayed in order to mark the hit zone

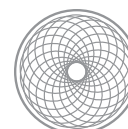
If a player misses, the spot will be left untouched.

Inputs are taken by ROW first followed by COLUMN

Press 1 to return to Main Menu



Before



Deploy your ships!

Deploy Carrier. Size is: 5

	B	A	T	T	L	E	S	H	I	P
	1	2	3	4	5	6				
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0

Deploy Carrier. Size is: 5

Type the coordinate for the front of the ship. ex. For(1,4) type 1 4 [Enter]

1 1

Deploy Carrier. Size is: 5

Type the coordinate for the rear of the ship. ex. For(1,4) type 1 4 [Enter]

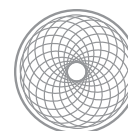
1 5

Deploy GunShip. Size is: 4

	B	A	T	T	L	E	S	H	I	P
	1	2	3	4	5	6				
1	1	1	1	1	1	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0

Deploy GunShip. Size is: 4

Type the coordinate for the front of the ship. ex. For(1,4) type 1 4 [Enter]



David Haro

After

Before

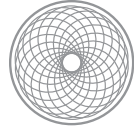
After

```
Deploy your ships!
Deploy Carrier. Size is: 5

Legend
-----
Hit Marker: 3
Ship Part: 1
```

	B	A	T	T	L	E	S	H	I	P
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

```
Deploy Carrier. Size is: 5
Type the coordinate for the front of the ship. ex. For(1,4) type 1 4 [Enter]
```



David Haro

Improvements since previous version (BattleShip_v2)

- * File: main.cpp
- * Author: David Haro
- * Battle Ship / Salvo Game
- * Last Update for BattleShip_V2: Dec 8th, 2013 12:10am
- * Current Version: BattleShip V2_5: Oct 26th, 2014 9:05pm
- *
- * Updates Since version (BattleShip_V2):
- * - Added Comments to function declarations
- * - Expanded battlefield board from 6x6 to 10x10 2D array
- * - Added player 2D Grids to be carried throughout the program
- * - Fixed issue where last row will error when checking for cross
- * - Added Legend to be displayed with player grid
- * - Structure added to hold Legend diagram
- * - Delete allocated memory before ending game
- * - Fixed issue where CPU would only deploy to a 6x6 playing field
- * - Squished other bugs!