**War of Ships**

**CSC 17A 43950**

**Nornubari Kanabolo**

**05/04/2015**

**Introduction**

In this game there is a naval war taking place between the British Empire (User) and Germany (Enemy) in 1839. They are fighting over control of the section of the Mediterranean Sea that has a less treacherous and more lucrative trade route, which can maximize only one of the country’s wealth. The program is similar to Battleship, and allows the user to decide the size of the board and how many ships should be used. The user chooses the spot on the board where he thinks the enemy ship is at by entering a row then a column within the bounds of the board. This continues until either the user or enemy has destroyed all the ships on their opponent’s board. The game then ends and displays the winner of the war.

**Summary**

Lines of code: 315

Comment Lines: 50

White spaces: 23

This project used most of the concepts and constructs we learned so far in this course. It uses a series of for loops, while, if-else-if statements and do while loops to control the moves, points, and winner of the game. The game uses structures for the board and players which had the components for the size of the board and the user and enemy ships, respectively. It also used functions with pass by references and dynamically allocated arrays to keep displaying the position of the two fleets’ ships during the war. It was difficult to figure out how to exactly nest some of the for loops since each one has a specific placement on the board.

**Pseudo Code**

**Game board and user input functions**

***Call begin function***

***Get board row and column size and number of ships from user***

***Print empty game board***

***Initialize i***

***For i<a.columns display empty space***

***Initialize i***

***Initialize j***

***For i<a.rows display empty space***

***i++***

***For j<a.columns display border section***

***j++***

***Initialize i***

***For i<a.columns display end of border section***

***i++***

***Dynamically allocate memory for arrays of the players***

***Call Player structure***

***Declare \*\*arr***

***Initialize variables***

***Set random number seed***

***Set spots where each ship will be blank***

***Do***

***Input tRow***

***while(tRow <= 0 ||tRow > b.rows)***

***Check if position is valid***

***Input tCol***

***while(tRow <= 0 ||tRow > b.columns)***

***check if position is valid***

***Initialize i***

***For i < b.ships***

***Do***

***Set random positions for enemy***

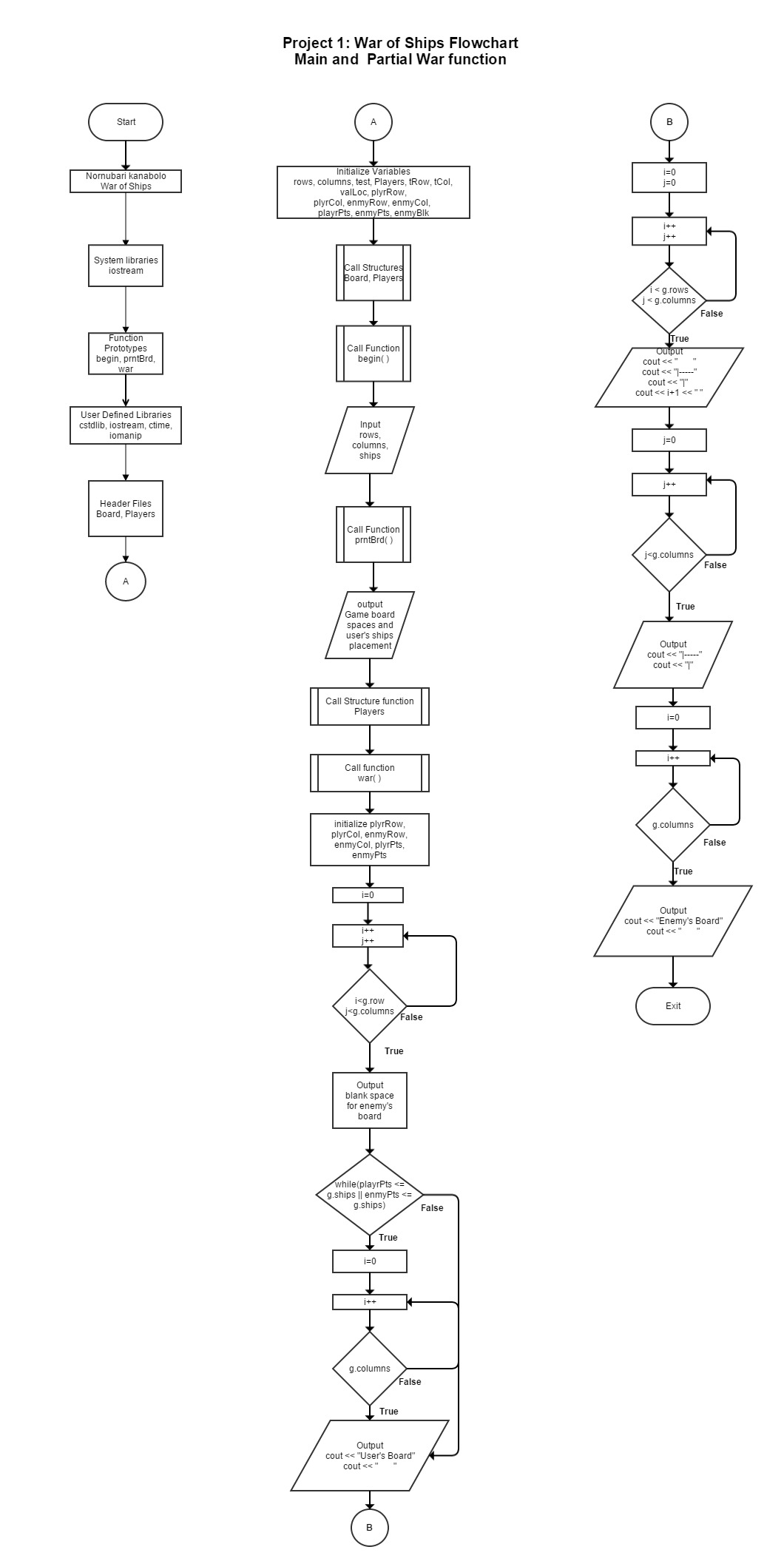
***while(valLoc == false)***

***Validate positions***

***Return arr***

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable Name | Description | Location |
| Const in | SIZE |  |  |
| int | hp | User health points | If statement where calculation of damage to user is done |
|  | ghp | Gremlin health points | If statement where calculation of damage to gremlin is done |
|  | atk | User attack power | User menu choices of attack |
|  | gatk | Gremlin attack power | Gremlin menu choices of attack |
|  | mag | User magic power | User menu choices of attack |
|  | gmag | Gremlin magic power | Gremlin menu choices of attack |
|  | def | User defense power | User menu choices of attack |
|  | gdef | Gremlin defense power | Gremlin menu choices of attack |
|  | hurt | User damage taken | If statement where amount of damage taken by user is calculated |
|  | ghurt | Gremlin damage taken | If statement where amount of damage taken by gremlin is calculated |
|  | choice | Move execution choice | Menu choices for user and gremlin |
|  | Start | Determines who makes the first move | The first if statement of program |
|  | usr\_score | Reference value for user score | Function prototypes |
|  | comp\_score | Reference value for gremlin | Function prototypes |
|  | value\_usr | Reference value for user counter | Function prototypes |
|  | value\_comp | Reference value for gremlin counter | Function prototypes |
|  | usr\_point | User score value | Variable declaration and function calls after user win |
|  | comp\_point | Gremlin score value | Variable declaration and function calls after gremlin win |
|  | value\_point | Counter value | Variable declaration and function calls after user or gremlin win |
|  | flag | Timer settings value | In timer function near end of program |
|  | strt | Start time | In cntDwn function near end of program |
|  | freeze | Time output increment | In cntDwn function near end of program |
|  | now | Current time | In timer function |
|  | timer | Stopwatch for user decision | Where user is prompted to take health potion |
|  | secs | Number of seconds to pause execution | In timer function |
|  | stp | Stop time | In cntDwn function |
|  | wait | Wait time before user decision | In cntDwn function |
| static int | strt | Remember start time | In timer function |
|  | secs | Time elapsed | In timer function |
| char | choice | Repeat program | After while loop at end of main() |
| string | map | Lines of the map file | Right after variable declaration |

|  |  |  |
| --- | --- | --- |
| Chapter | Constructs/Syntax | Location |
| 2 | Equality and relational operators  (&&, ||, ==,>=,>,<=,<,-,+) | Where health, damage given, and taken are compared or calculation |
|  | bool | In the functions for the repetition of the program and timer |
|  | If | Where the damage to user and gremlin are determined |
|  | If else | If the user starts the program or the monster does |
|  | While | While either the user or gremlins health points is greater than 0 or less than 1 |
|  | Do-while | Executes the menu at least once before checking if health for either is diminished |
|  | for | Loop to output the map |
| 3 | Switch | The menu options for the user and gremlin |
|  | & call by reference | Functions where scores are calculated and returned |
| 7 | Array | Displaying the map array at the start of the game |



**Program Code**

*/\**

*\* File: main.cpp*

*\* Author: Nornubari Kanabolo*

*\* CSC 5 Project 1*

*\*/*

**#include <iostream>**

**#include <ctime>**

**#include <cstdlib>**

**using** **namespace** std;

**int** **main**(**int** argc, **char\*\*** argv){

*//Declare variables*

**int** choice;

**int** hp, start, atk, def, mag, hurt, gatk, gdef, ghurt, gmag, ghp;

    atk **=** 12;*//user attack*

    def **=** 18;*//user defense*

    mag **=** 7;*//user magic*

    gatk **=** 14;*//gremlin attack*

    gdef **=** 18;*//gremlin defense*

    gmag **=** 7;*//gremlin magic*

*//Random seed that determines who starts*

    start **=** rand()**%**2**+**1;

    hp **=** rand()**%**60**+**100;*//health points user starts with*

    ghp **=** rand()**%**40**+**100;*//health points gremlin starts with*

*//User starts*

**if** (start **==** 1)

    {

        cout**<<**"You attack first and swiftly!";

        cout**<<**endl;

*//User Menu*

**while** (hp **>** 0 **||** ghp **>** 0) {

        cout**<<**"Choose what move you want to execute"**<<**endl;

        cout**<<**"1 - Strong Attack"**<<**endl;

        cout**<<**"2 - Magic Attack"**<<**endl;

        cout**<<**"3 - Defensive Move"**<<**endl;

**do**

        {

            cin**>>**choice;

        }**while**(choice**>**3 **||** choice**<**1);

**switch** (choice)

        {

**case** 1**:**

            atk **=** rand()**%**20**+**10;

            def **=** rand()**%**10**+**10;

            mag **=** rand()**%**5;

**break**;

**case** 2**:**

            atk **=** rand()**%**5**+**10;

            def **=** rand()**%**10**+**10;

            mag **=** rand()**%**15;

**break**;

**case** 3**:**

            atk **=** rand()**%**10**+**10;

            def **=** rand()**%**20**+**10;

            mag **=** rand()**%**5;

**break**;

         }

*//Gremlin Menu decides which move it executes*

        choice **=** rand()**%**3;

**switch** (choice)

        {

**case** 1**:**

            gatk **=** rand()**%**20**+**10;

            gdef **=** rand()**%**10**+**10;

            gmag **=** rand()**%**5;

**break**;

**case** 2**:**

            gatk **=** rand()**%**5**+**10;

            gdef **=** rand()**%**10**+**10;

            gmag **=** rand()**%**15;

**break**;

**case** 3**:**

            gatk **=** rand()**%**10**+**10;

            gdef **=** rand()**%**20**+**10;

            gmag **=** rand()**%**5;

**break**;

        }

*//Damage to gremlin*

        ghurt **=** (atk **-** gmag) **-** (gdef**/**atk);

**if** (ghurt **<** 0)

        {

            ghurt **=** 0;

        }

        ghp **=** ghp **-** ghurt;

        cout**<<**"You did "**<<**ghurt**<<**" damage to the gremlin!";

        cout**<<**endl;

*//If user defeats gremlin*

**if** (ghp **<** 1)

        {

            cout**<<**"You destroyed the gremlin! You are victorious with "**<<**hp**<<**" hp to spare.";

            cout**<<**endl;

**return** 0;

        }

        cout**<<**"The gremlin now has "**<<**ghp**<<**" hp left.";

        cout**<<**endl;

        hurt **=** (gatk **-** mag) **-** (def**/**gatk);

**if** (hurt **<** 0)

        {

            hurt **=** 0;

        }

        hp **=** hp **-** hurt;

        cout**<<**"The gremlin administered to you "**<<**hurt**<<**" damage.";

        cout**<<**endl;

*//If gremlin defeats user*

**if** (hp **<** 1)

        {

            cout**<<**"You have been defeated. The gremlin lives with "**<<**ghp**<<**" hp remaining.";

            cout**<<**endl;

**return** 0;

        }

        cout**<<**"You now have "**<<**hp**<<**" hp left.\n"**<<**endl;

        }

        }

*//The gremlin starts*

**else**

    {

        cout**<<**"Gremlin attacked first!"**<<**endl;

**while** (hp **>** 0 **||** ghp **>** 0) {

        choice **=** rand()**%**3;

**switch** (choice)

        {

**case** 1**:**

            gatk **=** rand()**%**20**+**10;

            gdef **=** rand()**%**10**+**10;

            gmag **=** rand()**%**5;

**break**;

**case** 2**:**

            gatk **=** rand()**%**5**+**10;

            gdef **=** rand()**%**10**+**10;

            gmag **=** rand()**%**15;

**break**;

**case** 3**:**

            gatk **=** rand()**%**10**+**10;

            gdef **=** rand()**%**20**+**10;

            gmag **=** rand()**%**5;

**break**;

        }

*//Gremlin does damage to user*

        hurt **=** (gatk **-** mag) **-** (def**/**gatk);

**if** (hurt **<** 0)

        {

            hurt **=** 0;

        }

        hp **=** hp **-** hurt;

        cout**<<**"The gremlin hit you for "**<<**hurt**<<**" damage.";

        cout**<<**endl;

*//If the gremlin kills the user*

**if** (hp **<** 1)

        {

            cout**<<**"You were killed. The gremlin still has "**<<**ghp**<<**" hp left.";

            cout**<<**endl;

**return** 0;

        }

        cout**<<**"You now have "**<<**hp**<<**" hp left.";

        cout**<<**endl;

*//Next move for user if not killed*

        cout**<<**"Choose what move you want to execute"**<<**endl;

        cout**<<**"1 - Strong Attack"**<<**endl;

        cout**<<**"2 - Magic Attack"**<<**endl;

        cout**<<**"3 - Defensive Move"**<<**endl;

**do**{cin**>>**choice;}**while**(choice**>**3 **||** choice**<**1);

**switch** (choice)

        {

**case** 1**:**

            atk **=** rand()**%**20**+**10;

            def **=** rand()**%**10**+**10;

            mag **=** rand()**%**5;

**break**;

**case** 2**:**

            atk **=** rand()**%**5**+**10;

            def **=** rand()**%**10**+**10;

            mag **=** rand()**%**15;

**break**;

**case** 3**:**

            atk **=** rand()**%**10**+**10;

            def **=** rand()**%**20**+**10;

            mag **=** rand()**%**5;

**break**;

        }

*//User hurts gremlin*

        ghurt **=** (atk **-** gmag) **-** (gdef**/**atk);

**if** (ghurt **<** 0)

        {

            ghurt **=** 0;

        }

        ghp **=** ghp **-** ghurt;

        cout**<<**"You did "**<<**ghurt**<<**" damage to the gremlin!";

        cout**<<**endl;

*//User wins*

**if** (ghp **<** 1)

        {

            cout**<<**"You destroyed the gremlin! You are victorious with "**<<**hp**<<**" hp remaining!";

            cout**<<**endl;

**return** 0;

        }

        cout**<<**"The gremlin now has "**<<**ghp**<<**" hp left."**<<**endl;

        cout**<<**endl;

        }

      }

}