Final Project: Jaws

# Table of contents

Doan Thach

CSC-5 section #45277

07/27/2016

# Introduction/summary --------------------------------------------------------pg.2

# Version updates------------------------------------------------------------------pg.3

# Iostream/cstdlib-----------------------------------------------------------------pg.3

# Ctime, iomanip, string, cmath------------------------------------------------pg.4

# Fstream/data types-------------------------------------------------------------pg.5

# Conditional statements, loops, functions--------------------------------pg.6

# Functions, arrays----------------------------------------------------------------pg.7

# Pseudocode-----------------------------------------------------------------------pg.8

# FlowChart-------------------------------------------------------------------------pg.9

# Source Code-----------------------------------------------------------------------pg.11

**Introduction**

Title: Jaws Game

Over the July 4th holiday, I saw a reshowing of the classic thriller movie “Jaws”. This inspired me to try to replicate the movie’s plot into a game that I will program.

Basically the program is a turn based game where the user has to battle a shark.

Every turn the program asks the user how they would like to attack the shark.

The harpoon does high damage, the pistol does mild damage, and the barrels increase attack power as well as inflict damage.

Once the player or the shark reaches 0 health points the game will end.

**Summary**

Project size: 600+ lines

The number of variables: 30

Coding this game took around 5 days.

I did not have any major problems coding this game. However I did have a lot of trouble with closing braces, attack values, and returning huge values for the Player’s health as well as the shark’s health.

**Version Updates**

V1 – Created and finished game

V2 – Added functions, fixed file I/O, added minigame with 2d array.

**Constructs and concepts used**

**Iostream library**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Frequency | Description | Location |
| Static\_cast | 1 | Statically cast as a different variable | 58 |
| cout | 143 | Output data | throughout |
| cin | 24 | Input data | throughout |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Frequency | Description | Location |
| Srand() | 1 | Random number seed | 58 |
| Rand() | 3 | Generates random # | 227, 348, 381 |
|  |  |  |  |

**cstdlib**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Frequency | Description | Location |
| time | 1 | Set current time | 58 |
|  |  |  |  |

**ctime library**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Frequency | Description | Location |
| Setprecision() | 3 | Format health | 235, 237, 403 |
| showpoint | 3 | Format health | 235, 237, 403 |
|  |  |  |  |

**Iomanip**

**String library**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Frequency | Description | Location |
| string | 3 | Declare variable | 61, 62, 433 |
|  |  |  |  |

**cmath**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Frequency | Description | Location |
| pow | 1 | For hooper’s Exponential damage | 367 |
|  |  |  |  |

**fstream library**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Frequency | Description | Location |
| Myfile.open | 1 | Open file | 85 |
| Myfile.close | 1 | Close file | 190 |
| ofstream | 1 | Declare variable | 84 |
|  |  |  |  |

**Data types:**

|  |  |  |
| --- | --- | --- |
| Name | Frequency | Location |
| Int | 6 | 68, 71, 77, 80, 81, 82 |
| Unsigned int | 4 | 63, 64, 65, 66 |
| Float | 7 | 67, 69, 70, 73, 74, 75, 76 |
| Char | 3 | 60, 78, 79 |
| Unsigned char | 1 | 72 |
| String | 4 | 61, 62 |
| bool | 3 | 51, 52, 53 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Name | Frequency | Location |
| if | 13 | 109, 112, 113, 208, 228, 244, 259, 277, 289, 319, 384, 390, 396 |
| If/else | 7 | 140, 263, 293, 312, 351, 446, 505 |
|  |  |  |

**Conditional statements:**

**Loops:**

|  |  |  |
| --- | --- | --- |
| Name | Frequency | Location |
| while | 5 | 99, 121, 193, 207, 331 |
| Do-while | 3 | 87, 105, 139 |
| for | 6 | 149, 366, 496, 497, 515, 516 |
|  |  |  |

**Functions:**

|  |  |  |
| --- | --- | --- |
| Return Type | Frequency | Location |
| void | 9 | 23, 24, 26, 32, 41, 42, |
| int | 6 | 27, 28, 30, 31, 47, 48, 49, 50 |
| float | 4 | 25, 29, 34, 38 |
|  |  |  |

**Functions:**

|  |  |  |
| --- | --- | --- |
| Name | Description | Location |
| Float battle | Pass by reference | 29, 372 |
| Float hoopAtk | Pass by value w/ defaulted parameter | 34, |
| Float battle ,  float hoopEmp | Multiple parameters | 29, 372  38, 364 |
|  |  |  |

**Arrays:**

|  |  |  |
| --- | --- | --- |
| Name | Description | Location |
| matrix | 2d array for tic-tac-toe minigame | 47 |
|  |  |  |
|  |  |  |

**Pseudocode**

Put in opening comments;

Bring in system libraries;

Introduce function prototypes;

Enter main, then immediately

set random number seed;

Declare all variables, initiate some

now and some later;

Open file that we will write data to;

Display menu;

Wait for user input

Display game

If not, display options or quit;

Introduce game background/characters;

Quint’s song trivia; if correct + attack power;

Prompt user if they want to play minigame

If yes, play minigame. If not, chum waters;

Shark attacks;

Determine if Hooper attacks based on remaining shark health;

If Hooper can attack, prompt user, calculate emp, then attack;

Player chooses attack style and attacks;

Calculate damage by randomly generating multiplier;

User’s damage done/damage taken is recorded;

Repeat shark’s/Hooper’s/User attacks until one dies;

User’s damage done/damage taken is recorded;

If Boat health is 0, transition to battle in the water;

Give user machete;

User decides to attack or hide;

If attack, damage shark and get bitten.

User’s damage done is recorded;

If hide, take mild damage;

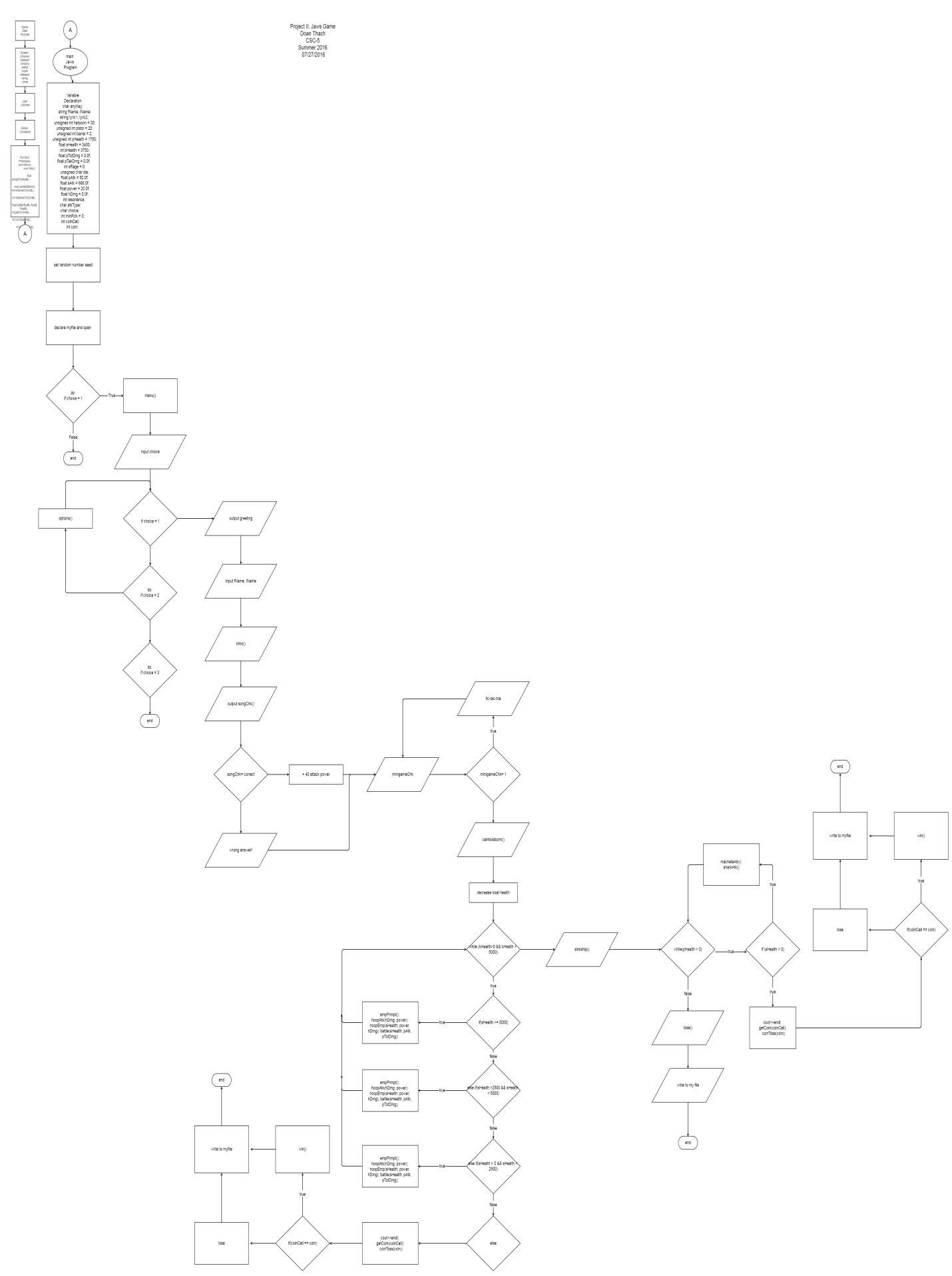
Damage taken is recorded;

Repeat battle in water until user or shark is dead;

Output winning/losing message;

**Flow Chart:**

The flowchart is too big, so I’ve attached it separately as well.



Source code:

/\*

\* File: main.cpp

\* Author: Doan Thach

\* Created on July 24, 2016, 8:30 PM

\* Purpose: Project II - Jaws/Shark Killer Game

\*/

//System Libraries

#include <iostream> //Input/Output

#include <iomanip> //Formatting

#include <cstdlib> //Random number Seed

#include <ctime> //Time for random seed

#include <cmath> //Math Library

#include <fstream> //File I/O

#include <string> //String Library

using namespace std;//Namespace of the System Libraries

//User Libraries

//Global Constants

//Function Prototypes

void menu(); //Menu

void intro(); //Intro

float songChk(float&); //Song Check Game

void calmB4Storm();

int mGameChk(int&); //Mini Game check

int mGameChk2(int&); //Mini Game continue check

float battle(float&, float&, float&); //Battle function

int getCoin(int&); //Get player coin side function

int coinToss(int&); //Coin Toss function

void empPrmpt(); //Hooper Attack prompt

//Hooper's attack damage calculation function

float hoopAtk(float& hDmg, float power, int resonance=10){

hDmg = power + resonance;

return hDmg;

}

float hoopEmp(float&, float&, float); //Hooper's Attack function

char check(void);

void init\_matrix(void);

void get\_player\_move(void);

void get\_computer\_move(void);

void disp\_matrix(void);

void reset();

//--------------------initial matrix declaration--------------------------------

int matrix[3][3];

int row, column, diagonal;

int empty\_r(int c);

int empty\_c(int r);

bool vulnerable(int mat[3][3]);

bool winnable(int mat[3][3]);

bool finished(int mat[3][3]);

//-----------------------------------------------------------------------------

//Execution Begins Here!

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

//Set the random number seed

srand(static\_cast<unsigned int>(time(0)));

//Declare Variables

char anyKey;

string fName, lName;

string lyric1, lyric2;

unsigned int harpoon = 30;

unsigned int pistol = 20;

unsigned int barrel = 2;

unsigned int pHealth = 1750;

float sHealth = 3400;

int bHealth = 3750;

float pTotDmg = 0.0f;

float pTakDmg = 0.0f;

int sRage = 0;

unsigned char die;

float pAtk = 50.0f;

float sAtk = 666.0f;

float power = 20.0f;

float hDmg = 0.0f;

int resonance;

char atkType; //Type of attack

char choice; //menu choice

int minPick = 0; //mini game choice

int coinCall; //Side of coin called by player

int coin; //Outcome of coin

//Open files and Input Data

ofstream myfile;

myfile.open ("Your Game Stats.txt");

//Input Data

do{

menu();

cin>>choice;

//Process the Data and Display Results

switch(choice){

case '1':

cout<<"You have chosen to play!"<<endl<<endl;

cout<<"Please enter your first and last name:"<<endl;

cin>>fName>>lName;

intro();

songChk(pAtk);

mGameChk(minPick);

while(minPick == 1){

char done;

cout<<"You vs. Hooper!"<<endl;

cout<<"You will be playing against Hooper as 'X'"<<endl;

done = ' ';

init\_matrix();

do {

disp\_matrix();

get\_player\_move();

done = check(); /\* check winner \*/

if(done!= ' ') break; /\* if winner found...\*/

get\_computer\_move();

done = check(); /\* check for winner again \*/

if(done!= ' ') break; /\* if winner found...\*/

if (finished(matrix)) //we don't have a winner and there are no open spaces.

{

disp\_matrix();

cout << endl;

cout<<"draw\n";

break;

}

reset();

} while(done== ' ');

if(done=='X')

cout<<"You won! However Hooper seems drunk...\n";

else

cout<<"Hooper is so drunk yet he can still beat you..."<<endl;

disp\_matrix(); /\* show final positions \*/

mGameChk2(minPick);

}

calmB4Storm();

//-------------------------Battle-------------------------------

cout<<"Your boat has been attacked by the shark!"<<endl;

bHealth = bHealth-sAtk;

cout<<"Your boat takes "<<sAtk<<" damage!"<<endl;

cout<<"Boat hull strength: "<<bHealth<<endl;

cout<<"Enter any key to continue"<<endl;

cin>>anyKey;

cout<<"After recovering from the initial shock, you"

<<" decide to attack the shark!"<<endl;

do{

if(sHealth >= 5000){

cout<<"The shark is still full of vigor!"<<endl;

cout<<"The shark attacks your boat again!"<<endl;

bHealth = bHealth-sAtk;

cout<<"Your boat takes "<<sAtk<<" damage!"<<endl;

cout<<"Boat hull strength: "<<bHealth<<endl;

cout<<"You decide to attack the shark!"<<endl;

cout<<"Enter any key to continue"<<endl;

cin>>anyKey;

for(int i = 1; i < 2; i++){

empPrmpt();

hoopAtk(hDmg, power);

hoopEmp(sHealth, power, hDmg);

}

battle(sHealth, pAtk, pTotDmg);

}else if(sHealth >2500 && sHealth < 5000){

cout<<"The shark is showing signs of fatigue"<<endl;

cout<<"The shark attacks your boat again!"<<endl;

bHealth = bHealth-sAtk;

cout<<"Your boat takes "<<sAtk<<" damage!"<<endl;

cout<<"Boat hull strength: "<<bHealth<<endl;

cout<<"You decide to attack the shark!"<<endl;

cout<<"Enter any key to continue"<<endl;

cin>>anyKey;

battle(sHealth, pAtk, pTotDmg);

}else if(sHealth > 0 && sHealth < 2500){

cout<<"The shark is nearing death!"<<endl;

cout<<"The shark attacks your boat again!"<<endl;

bHealth = bHealth-sAtk;

cout<<"Your boat takes "<<sAtk<<" damage!"<<endl;

cout<<"Boat hull strength: "<<bHealth<<endl;

cout<<"You decide to attack the shark!"<<endl;

cout<<"Enter any key to continue"<<endl;

cin>>anyKey;

battle(sHealth, pAtk, pTotDmg);

}else{

cout<<endl;

getCoin(coinCall);

coinToss(coin);

if(coinCall == coin){

cout<<"You beat Quint! You earn $22,000!"<<endl;

}else{

cout<<"You lost to Quint! You earn $11,000!"<<endl;

}

myfile<<"Player name: "<<fName<<" "<<lName<<"."<<endl;

myfile<<"\n"<<endl;

myfile <<" The total damage you have done to the shark is: \n"

<<pTotDmg<<"."<<endl;

myfile <<"\n The total damage you have sustained from the shark"

<<" is: "<<pTakDmg<<"."<<endl;

myfile.close();

return 0;

}

}while(bHealth>0 && bHealth < 5000);

//-----------------------Battle end-----------------------------

//---------------------------Battle in water--------------------

if(bHealth <= 0 && pHealth > 0 && sHealth > 0){

cout<<endl;

cout<<"Your boat has been destroyed and you along with your"

<<" friends have fallen into the water!"<<endl;

cout<<"You watch as Quint gets eaten alive! You search for "

<<"Hooper, but he is no where to be found!"<<endl;

cout<<endl<<"The odds are definitely against you!"<<endl;

cout<<"Luckily, you were able to find a machete lodged into "

<<"debris from the ship"<<endl;

cout<<"Your health: "<<pHealth<<endl;

cout<<"Shark's Health: "<<sHealth<<endl;

while(pHealth >= 0){

if(pHealth <= 0 || pHealth > 5000){

cout<<endl;

cout<<"The shark catches you off guard and devours you!"

<<endl<<"!!!You are DEAD!!!"<<endl;

cout<<"The credits roll and your soiled swim trunks"

<<" float up to the surface of the water..."<<endl;

myfile<<"Player name: "<<fName<<" "<<lName<<"."<<endl;

myfile<<"\n"<<endl;

myfile <<" The total damage you have done to the shark is: \n"

<<pTotDmg<<"."<<endl;

myfile <<"\n The total damage you have sustained from the shark"

<<" is: "<<pTakDmg<<"."<<endl;

myfile.close();

return 0;

}

cout<<"What do you want to do?"<<endl;

cout<<"(1) Attack"<<endl;

cout<<"(2) Try to Hide"<<endl;

cin>>atkType;

die=rand()%6+1; //Random number [1,6]

if(atkType=='1'){

sHealth = sHealth-(pAtk+40)+(pAtk\*die/8);

cout<<"You attack the shark with the machete and do "

<<(pAtk+35)+(pAtk\*die/8)<<" damage!"<<endl;

cout<<"The shark swims by you and bites you!"<<endl;

cout<<"You bleed for 250 damage!"<<endl;

pHealth = pHealth - 250;

cout<<"Shark Health: "<<setprecision(4)<<showpoint

<<sHealth<<endl;

cout<<"Your health: "<<setprecision(4)<<showpoint

<<pHealth-250<<endl;

cout<<"The shark smells your blood and narrows its "

<<"search area!"<<endl;

pTotDmg += (pAtk+40)+(pAtk\*die/8);

pTakDmg += 250;

}

if(pHealth <= 0 || pHealth > 5000){

cout<<endl;

cout<<"The shark catches you off guard and devours you!"

<<endl<<"!!!You are DEAD!!!"<<endl;

cout<<"The credits roll and your soiled swim trunks"

<<" float up to the surface of the water..."<<endl;

myfile<<"Player name: "<<fName<<" "<<lName<<"."<<endl;

myfile<<"\n"<<endl;

myfile <<" The total damage you have done to the shark is: \n"

<<pTotDmg<<"."<<endl;

myfile <<"\n The total damage you have sustained from the shark"

<<" is: "<<pTakDmg<<"."<<endl;

myfile.close();

return 0;

}

if(sHealth<=0){

cout<<endl;

getCoin(coinCall);

coinToss(coin);

if(coin == coinCall){

cout<<"You beat Hooper! You earn $22,000!"<<endl;

}else{

cout<<"You lost to Hooper! You earn $11,000!"<<endl;

}

myfile<<"Player name: "<<fName<<" "<<lName<<"."<<endl;

myfile<<"\n"<<endl;

myfile <<" The total damage you have done to the shark is: \n"

<<pTotDmg<<"."<<endl;

myfile <<"\n The total damage you have sustained from the shark"

<<" is: "<<pTakDmg<<"."<<endl;

myfile.close();

return 0;

}

if(atkType=='2'){

cout<<"You try to hide from the shark inside some"

<<"debris."<<endl;

cout<<"The debris hurts you a bit"<<endl;

pHealth = pHealth - 75;

cout<<"Your health: "<<pHealth;

cout<<"The shark smells your blood and narrows its "

<<"search area!"<<endl;

pTakDmg += 75;

}

}

}

if(sHealth <=0 ){

cout<<endl;

getCoin(coinCall);

coinToss(coin);

if(coin == coinCall){

cout<<"You beat Hooper! You earn $22,000!"<<endl;

}else{

cout<<"You lost to Hooper! You earn $11,000!"<<endl;

}

myfile<<"Player name: "<<fName<<" "<<lName<<"."<<endl;

myfile<<"\n"<<endl;

myfile <<" The total damage you have done to the shark is: \n"

<<pTotDmg<<"."<<endl;

myfile <<"\n The total damage you have sustained from the shark"

<<" is: "<<pTakDmg<<"."<<endl;

myfile.close();

}

break;

case '2':

char opt1;

char opt2;

cout<<"You have chosen options:"<<endl;

cout<<"Do you want to change your initial health? (y/n)"<<endl;

cin>>opt1;

if(opt1=='y'||opt1=='Y'){

cout<<"Please enter the amount of initial health you want"<<endl;

cin>>pHealth;

}else{

cout<<"Do you want to change your initial attack power?"<<endl;

cin>>opt2;

if(opt2=='y'||opt2=='Y'){

cout<<"Please enter the initial attack power you want"<<endl;

cin>>pAtk;

}

break;

case '3':

cout<<"Goodbye!"<<endl;

return 0;

break;

default:cout<<"Not option in menu"<<endl;

}

}

}while(choice>='1'&&choice<='3');

//Exit Stage Right!

return 0;

}

int getCoin(int& coinCall){

cout<<"!!!CONGRATULATIONS!!!"<<endl;

cout<<"You have successfully killed the shark!"<<endl;

cout<<"The $10,000 bounty is yours!"<<endl;

cout<<"Hooper: 'Don't forget about my share!'"<<endl;

cout<<"You and Hooper decide to flip a coin to decide"

<<" who will get two-thirds of the bounty!"<<endl;

cout<<"Call the side of the coin you want:"<<endl;

cout<<"(1)Heads"<<endl;

cout<<"(2)Tails"<<endl;

cin>>coinCall;

return coinCall;

}

int coinToss(int& coin){

unsigned char die;

die=rand()%6+1; //Random number [1,6]

if(die%2==0){

int coin = 1;

cout<<"The coin came up heads!"<<endl;

}else{

int coin = 2;

cout<<"The coin came up tails!"<<endl;

}

return coin;

}

void empPrmpt(){

char anyKey;

cout<<"Hooper EMPs the shark!"<<endl;

}

float hoopEmp(float& sHealth, float& power, float hDmg){

sHealth = sHealth - hDmg;

for(int i = 1; i<=3; i++){

power = power+((power\*i)/2);

}

cout<<"Hooper hits the shark for "<<hDmg<<" damage!"<<endl;

return power;

}

float battle(float& sHealth, float& pAtk, float& pTotDmg){

unsigned int harpoon = 50;

unsigned int pistol = 40;

unsigned int barrel = 2;

unsigned char die;

char atkType;

cout<<"How do you want to attack the shark?"<<endl;

cout<<"(1) Harpoon"<<endl;

cout<<"(2) Pistol"<<endl;

cout<<"(3) Attach Barrel"<<endl;

cin>>atkType;

die=rand()%6+1; //Random number [1,6]

if(atkType=='1'){

sHealth = sHealth-(pAtk+harpoon)+(pAtk\*die/10);

cout<<"You attack the shark with a harpoon and do "

<<(pAtk+harpoon)+(pAtk\*die/10)<<" damage!"<<endl;

pTotDmg += (pAtk+harpoon)+(pAtk\*die/10);

}

if(atkType=='2'){

sHealth = sHealth-(pAtk+pistol)+(pAtk\*die/10);

cout<<"You attack the shark with a pistol and do "

<<(pAtk+pistol)+(pAtk\*die/10)<<" damage!"<<endl;

pTotDmg += (pAtk+pistol)+(pAtk\*die/10);

}

if(atkType=='3'){

pAtk += 20;

sHealth = sHealth-(pAtk+(pAtk\*barrel)+(pAtk\*die/10));

cout<<"You attach the shark with a barrel and do "

<<pAtk+(pAtk\*barrel)+(pAtk\*die/10)<<" damage!"<<endl;

pTotDmg += (pAtk+(pAtk\*barrel)+(pAtk\*die/10));

}

cout<<"Shark Health: "<<setprecision(4)<<showpoint<<sHealth<<endl;

return sHealth;

}

void calmB4Storm(){

char anyKey;

cout<<"You decide to chum the waters for a while..."<<endl;

cout<<"Enter any key to continue"<<endl;

cin>>anyKey;

cout<<"Nothing seems to be happening so you take a nap"<<endl;

cout<<"Enter any key to continue"<<endl;

cin>>anyKey;

cout<<"ZZZzzz ZZzz ZZz zzz ..."<<endl;

cout<<"Enter any key to continue"<<endl;

cin>>anyKey;

}

int mGameChk2(int& minPick){

cout<<"Do you want to play against Hooper again?"<<endl;

cout<<"Please enter: (1) to play with Hooper"<<endl;

cout<<" (2) to chum the waters"<<endl;

cin>>minPick;

}

int mGameChk(int& minPick){

cout<<"Hooper has become bored and asks if you would like to "

<<"play a game of Tic-Tac-Toe with him"<<endl;

cout<<"Do you accept?"<<endl;

cout<<"Please enter: (1) to play with Hooper"<<endl;

cout<<" (2) to chum the waters"<<endl;

cin>>minPick;

}

float songChk (float& pAtk){

string lyric1, lyric2;

cout<<" Along the way, Quint starts to sing a sailor's song"

<<" known to boost the strength of whomever sings it"<<endl;

cout<<"Quint: 'Farewell and adieu to you fair Spanish ladies "<<endl;

cout<<" Farewell and adieu you ladies of (1)\_\_\_\_\_. "<<endl;

cout<<" For we received orders for to sail back to Boston"<<endl;

cout<<" And soon never more will we see you (2)\_\_\_\_\_.'"<<endl;

cout<<"Please finish the song to gain additional attack power"<<endl;

cout<<"Enter the first missing lyric"<<endl;

cin>>lyric1;

cout<<"Enter the second missing lyric"<<endl;

cin>>lyric2;

if((lyric1=="spain" || lyric1=="Spain") && lyric2=="again"){

pAtk +=40;

cout<<"You gain +40 to your attack power!"<<endl<<endl;

}else{

cout<<"Quint: That's not how the song goes!"<<endl<<endl;

}

return pAtk;

}

void intro(){

char anyKey;

cout<<" For the past week, the summer colony of Amity Island"

<<" has been plagued with attacks from a rogue "<<endl;

cout<<"great white shark. The town is desperate to get rid"

<<" of the shark by July 4th, the most profitable"<<endl;

cout<<"time for the island. The town is offering a $33,000 "

<<"bounty for the shark dead or alive."<<endl;

cout<<" Being a poor university student you find this offer"

<<" too good to turn down, so you pack your bags and "<<endl;

cout<<"head to Amity Island to become a shark hunter."<<endl<<endl;

cout<<"Press any key to continue"<<endl;

cin>>anyKey;

cout<<" When you arrive on Amity Island, a local shark hunter"

<<" by the name of Quint offers to help you hunt the"<<endl;

cout<<"shark for a share of the bounty...Seeing as there are"

<<" no other fishermen willing to help you, you "<<endl;

cout<<"begrudgingly agree."<<endl<<endl;

cout<<"Press any key to continue"<<endl;

cin>>anyKey;

cout<<"As you and Quint, load up the boat with equipment, you notice a "

<<"red-headed man trying to get your attention from the docks..."<<endl;

cout<<"It seems that the local authorities have asked a shark expert "

<<"by the name of Hooper to help anyone planning on hunting the shark."<<endl;

cout<<"Hooper:'Hey I'm going to help you kill that shark!"<<endl;

cout<<" Those other fishermen are nothing but drunken idiots with dynamite!"<<endl;

cout<<"Press any key to continue"<<endl;

cin>>anyKey;

cout<<"Armed with harpoons, barrels, and a pistol. You"

<<", Quint, and Hooper set sail to find and kill the shark!"<<endl<<endl;

}

void menu(){

cout<<"Welcome to Shark Hunter"<<endl;

cout<<"What do you want to do?"<<endl;

cout<<"Input 1 to play"<<endl;

cout<<"Input 2 for options"<<endl;

cout<<"Input 3 to quit"<<endl;

cout<<endl;

}

void init\_matrix(void) //matrix intitialisation

{

int i, j;

for(i=0; i<3; i++)

for(j=0; j<3; j++) matrix[i][j] = ' ';

}

void get\_player\_move(void) //call function for player input

{

int x, y;

cout<<"Enter X,Y coordinates for your move: ";

cin >> x >> y;

x--; y--;

if(matrix[x][y]!= ' '){

cout<<"Invalid move, try again.\n";

get\_player\_move();

}

else matrix[x][y] = 'X';

}

void get\_computer\_move(void) //AI move input

{

char temp;

int test[3][3], temporary[3][3];

for (int i = 0; i < 3; ++i)

for (int j = 0; j < 3; ++j)

{ test[i][j] = matrix[i][j]; temporary[i][j] = matrix[i][j]; }

int i, j;

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

if (matrix[i][j] == ' ')

{

temp = matrix[i][j];

if (winnable(test))

{

if (row > 0)

{

if (matrix[row-1][empty\_c(row-1)] != 'X')

matrix[row-1][empty\_c(row-1)] = 'O';

return;

}

if (column > 0)

{

if (matrix[empty\_r(column-1)][column-1] != 'X')

matrix[empty\_r(column-1)][column-1] = 'O';

return;

}

if (diagonal == 1)

{

if ( matrix[0][0] !='X' && matrix[1][1] != 'X' && matrix[2][2] != 'X')

{matrix[0][0] = 'O'; matrix[1][1] = 'O'; matrix[2][2] = 'O';}

return;

}

else

{

if (matrix[0][2] != 'X' && matrix[1][1] != 'X' && matrix[2][0] != 'X')

{matrix[0][2] = 'O'; matrix[1][1] = 'O'; matrix[2][0] ='O';}

return;

}

return;

//break;

}

test[i][j] = temp;

}

if (matrix[i][j]==' ')

{

temp = matrix[i][j];

if (winnable(test))

{

if (row > 0)

{

if (matrix[row-1][empty\_c(row-1)] != 'X')

matrix[row-1][empty\_c(row-1)] = 'O';

return;

}

if (column > 0)

{

if (matrix[empty\_r(column-1)][column-1] != 'X')

matrix[empty\_r(column-1)][column-1] = 'O';

return;

}

if (diagonal == 1)

{

if ( matrix[0][0] !='X' && matrix[1][1] != 'X' && matrix[2][2] != 'X')

{matrix[0][0] = 'O'; matrix[1][1] = 'O'; matrix[2][2] = 'O';}

return;

}

else

{

if (matrix[0][2] != 'X' && matrix[1][1] != 'X' && matrix[2][0] != 'X')

{matrix[0][2] = 'O'; matrix[1][1] = 'O'; matrix[2][0] = 'O';}

return;

}

return;

}

test[i][j] = temp;

}

}

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

if (matrix[i][j]==' ')

{

temp = matrix[i][j];

test[i][j] = 'O';

if (!vulnerable(test))

break;

test[i][j] = temp;

}

if (matrix[i][j]==' ')

{

temp = matrix[i][j];

test[i][j] = 'O';

if (!vulnerable(test))

break;

test[i][j] = temp;

}

}

if (column > 0)

{

if (matrix[empty\_r(column-1)][column-1] !='X')

matrix[empty\_r(column-1)][column-1] = 'O';

return;

}

if (row > 0)

{

if (matrix[row-1][empty\_c(row-1)] !='X')

matrix[row-1][empty\_c(row-1)] = 'O';

return;

}

if (diagonal == 1 )

{

if (matrix[0][0] != 'X')

matrix[0][0] = 'O';

if (matrix[1][1] != 'X')

matrix[1][1] = 'O';

if (matrix[2][2] != 'X')

matrix[2][2] = 'O';

return;

}

if (diagonal == 2 )

{

if (matrix[0][2] != 'X')

matrix[0][2] = 'O';

if (matrix[1][1] != 'X')

matrix[1][1] = 'O';

if (matrix[2][0] != 'X')

matrix[2][0] = 'O';

return;

}

if (matrix[i][j] != 'X')

matrix[i][j] = 'O';

}

void disp\_matrix(void) //matrix display

{

int t;

for(t=0; t<3; t++)

{

cout << " " << char(matrix[t][0]) << " | " << char(matrix[t][1]) << " | " << char(matrix[t][2]);

if(t!=2)

cout << "\n---|---|---\n";

}

cout << endl;

}

char check(void) //used for identifying winner

{

int i;

for(i=0; i<3; i++) /\* check rows \*/

if(matrix[i][0]==matrix[i][1] &&

matrix[i][0]==matrix[i][2]) return matrix[i][0];

for(i=0; i<3; i++) /\* check columns \*/

if(matrix[0][i]==matrix[1][i] &&

matrix[0][i]==matrix[2][i]) return matrix[0][i];

/\* test diagonals \*/

if(matrix[0][0]==matrix[1][1] &&

matrix[1][1]==matrix[2][2])

return matrix[0][0];

if(matrix[0][2]==matrix[1][1] &&

matrix[1][1]==matrix[2][0])

return matrix[0][2];

return ' ';

}

bool vulnerable(int mat[3][3])

{

char a, b, c, d, e, f, g, h, i;

a = mat[0][0], b = mat[0][1], c = mat[0][2];

d = mat[1][0], e = mat[1][1], f = mat[1][2];

g = mat[2][0], h = mat[2][1], i = mat[2][2];

if ( (a=='X' && b=='X' && c!='O') || (b=='X' && c=='X' && a!='O') || (a=='X' && c=='X' && b!='O') )

{ row = 1; return true;}

if ( (d=='X' && e=='X' && f!='O') || (e=='X' && f=='X' && d!='O') || (d=='X' && f=='X' && e!='O') )

{ row = 2; return true;}

if ( (g=='X' && h=='X' && i!='O') || (h=='X' && i=='X' && g!='O') || (g=='X' && i=='X' && h!='O') )

{ row = 3; return true;}

if ( (a=='X' && d=='X' && g!='O') || (d=='X' && g=='X' && a!='O') || (a=='X' && g=='X' && d!='O') )

{ column = 1; return true;}

if ( (b=='X' && e=='X' && h!='O') || (e=='X' && h=='X' && b!='O') || (b=='X' && h=='X' && e!='O') )

{ column = 2; return true;}

if ( (c=='X' && f=='X' && i!='O') || (f=='X' && i=='X' && c!='O') || (c=='X' && i=='X' && f!='O') )

{ column = 3; return true;}

if ( (a=='X' && e=='X' && i!='O') || (e=='X' && i=='X' && a!='O') || (a=='X' && i=='X' && e!='O') )

{ diagonal = 1; return true; }

if ( (g=='X' && e=='X' && c!='O') || (e=='X' && c=='X' && g!='O') || (g=='X' && c=='X' && e!='O') )

{ diagonal = 2; return true; }

return false;

}

bool winnable(int mat[3][3])

{

char a, b, c, d, e, f, g, h, i;

a = mat[0][0], b = mat[0][1], c = mat[0][2];

d = mat[1][0], e = mat[1][1], f = mat[1][2];

g = mat[2][0], h = mat[2][1], i = mat[2][2];

if ( (a=='O' && b=='O' && c!='X') || (b=='O' && c=='O' && a!='X') || (a=='O' && c=='O' && b!='X') )

{

row = 1;

return true;

}

if ( (d=='O' && e=='O' && f!='X') || (e=='O' && f=='O' && d!='X') || (d=='O' && f=='O' && e!='X') )

{

row = 2;

return true;

}

if ( (g=='O' && h=='O' && i!='X') || (h=='O' && i=='O' && g!='X') || (g=='O' && i=='O' && h!='X') )

{

row = 3;

return true;

}

if ( (a=='O' && d=='O' && g!='X') || (d=='O' && g=='O' && a!='X') || (a=='O' && g=='O' && d!='X') )

{

column = 1;

return true;

}

if ( (b=='O' && e=='O' && h!='X') || (e=='O' && h=='O' && b!='X') || (b=='O' && h=='O' && e!='X') )

{

column = 2;

return true;

}

if ( (c=='O' && f=='O' && i!='X') || (f=='O' && i=='O' && c!='X') || (c=='O' && i=='O' && f!='X') )

{

column = 3;

return true;

}

if ( (a=='O' && e=='O' && i!='X') || (e=='O' && i=='O' && a!='X') || (a=='O' && i=='O' && e!='X') )

{

diagonal = 1;

return true;

}

if ( (g=='O' && e=='O' && c!='X') || (e=='O' && c=='O' && g!='X') || (g=='O' && c=='O' && e!='X') )

{

diagonal = 2;

return true;

}

return false;

}

bool finished(int mat[3][3])

{

for (int i = 0; i < 3; ++i)

{

for (int j = 0; j < 3; ++j)

{

if (mat[i][j] == ' ')

return false;

}

}

return true;

}

int empty\_c(int r)

{

for (int i = 0; i < 3; ++i)

{

if (matrix[r][i] == ' ')

return i;

}

}

int empty\_r(int c){

for (int i = 0; i < 3; ++i)

{

if (matrix[i][c] == ' ')

return i;

}

}

void reset(){

row = 0;

column = 0;

diagonal = 0;

}