#include <stdio.h>

#include <stdlib.h>

#include <time.h>

//Franklyn Gonzalez, last edited 02/18/2019

/\*Cheatsheet is used for the programmer to show the hidden ships in the game. (Not Included) \*/

void welcomeScreen();

void emptymap(int map[][10],int noOfships);

void instructions();

void stage(int map[][10], int noOfships);

void bomb(int row1, int col1, int map[][10], int noOfships);

//void cheatsheet(int map[][10]);

int main()

{//start main

int level, start, bombrow, bombcol, noOfships;

int map[10][10] = { 0 };

welcomeScreen(); //Call welcome

scanf\_s("%d", &start);

while ((start < 0) || (start > 1))

{//start while

printf("Please enter a valid input: \n");

scanf\_s("%d", &start);

}//while end

system("cls"); //used to clean the loading screen to start loading the map on its own

while (start== 1)

{//start while

instructions(); // Call instruction

scanf\_s("%d", &level);

while ((level != 1))

{//start while

printf("Please enter the correct level: \n");

scanf\_s("%d", &level);

}//end while

switch (level)

{//start of switch case

case 1:

noOfships = 5;

break;

}//end of switch case

system("cls"); //used to clean the loading screen to start loading the map on its own

stage(map, noOfships); //To pass noOfships value to stage

//cheatsheet(map); //To show cheat sheet by displaying ships (Not Included)

emptymap(map,noOfships); //Pass value to emptymap

printf("\nEnter (row 'spacebar' col) to bomb:\n");

scanf\_s("%d%d", &bombrow, &bombcol);

while ((bombrow <= 0) || (bombrow > 10) || (bombcol <= 0) || (bombcol > 10))

{//start while

printf("\nEnter (row 'spacebar' col) to bomb:\n");

scanf\_s("%d%d", &bombrow, &bombcol);

}//while end

bomb(bombrow, bombcol, map, noOfships); //Pass value bombrow and bombcol to bomb

printf("Enter '1' to continue or '0' to quit game\n");

scanf\_s("%d", &start);

while ((start < 0) || (start > 1)) //To check if user wants to replay the game

{//start while

printf("Please enter a valid input: \n");

scanf\_s("%d", &start);

}//while end

}//while end

system("pause");

}//end of main function

void welcomeScreen() //Function to print Welcome screen.

{//Start of welcomescreen

printf("======================================================================\n");

printf(" XXXXX XXXX XXXXXX XXXXXX XX XXXXXX XXXXX XX XX XX XXXX\n");

printf(" XX XX XX XX XX XX XX XX XX XX XX XX XX XX\n");

printf(" XXXXX XX XX XX XX XX XXXX XXXX XXXXXX XX XXXX\n");

printf(" XX XX XXXXXX XX XX XX XX XX XX XX XX XX\n");

printf(" XXXXX XX XX XX XX XXXXXX XXXXXX XXXXX XX XX XX XX\n");

printf("======================================================================\n");

printf("\n");

printf("Enter '1' to continue or '0' to quit game\n");

}//function-end

void instructions() //Function to print instruction screen

{//Start of intruction-fucntion

printf("There is only 1 level in this game: Beginner.\n");

printf("You will have to enter the rows and col to DESTROY the hidden ships.\n");

printf("To win the game you will have 10 attempts to sink 5 ships.\n");

printf("There will be 5 hidden ships for Beginner level.\n");

printf("Enter 1: Beginner level.\n");

}//End of instruction

void emptymap(int map[][10],int noOfships) //Function to show map

{// Start of emptymap

int rows, col;

printf("%13d\n", 1);

printf(" ABCDEFGHIJ");

printf("\n");

for (rows = 1; rows <= 10; rows++)

{

printf("%2d ", rows);

for (col = 0; col < 10; col++)

{

if ((map[rows - 1][col] != -1)&& (map[rows - 1][col] != -2)) // used to setup the gameboard

{

printf("#");

}//end if

else if (map[rows-1][col] == -1) // ship has been hit!

{

printf("H");

}//end if

else if (map[rows-1][col] == -2)

{

printf(" ");

}//end if

}//loop-end

printf("\n");

}//loop-end

}// End of emptymap-function

void stage(int map[][10], int noOfships) //Function to generate number of ships

{//Start of stage-function

int x, y, ship = 1;

srand((time)NULL); // to prevent sequence repetition

while (ship <= noOfships)

{

x = rand() % 10;

y = rand() % 10;

if ((map[x][y] == 0) && (map[x][y + 4] == 0) && ((y + 4) < 10)) //values have been set within range

{

map[x][y + 2] = ship;

map[x][y + 3] = ship;

map[x][y + 4] = ship;

map[x][y + 5] = ship;

ship++;

}//end-if

}//while-end

}//End of stage-function

void bomb(int row1, int col1, int map[][10],int noOfships) //Function to shoot the hidden ships

{//Start of bomb-function

system("cls");

int point = 0;

int attempt = 10; // adjust number of attempts to change difficulty of the game

int count = 0;

int x, y, coordinate;

while ((attempt > 0) && (count != 5))

{

attempt = attempt - 1;

if ((map[row1 - 1][col1 - 1] >= 1) && (map[row1 - 1][col1 - 1] <= noOfships))

{

coordinate = map[row1 - 1][col1 - 1];

for (x = 0; x <= 9; x++)

{

for (y = 0; y <= 9; y++)

{

if (map[x][y] == coordinate)

{

map[x][y] = -1;

}//End-if

}//Loop-end

}//Loop-end

count++;

}//End-if

else if ((map[row1 - 1][col1 - 1] == 0))

{

printf("Try again.\n");

map[row1 - 1][col1 - 1] = -2;

}//End-if

printf("You have hit %d of 5 ships.\n", count);

printf("You still have %d attempt.\n", attempt);

emptymap(map,noOfships); //To call emptymap

printf("Enter (row 'spacebar' col) to bomb:");

scanf\_s("%d%d", &row1, &col1);

while ((row1 <= 0) || (row1 > 10) || (col1 <= 0) || (col1 > 10))

{

printf("\nEnter (row 'spacebar' col) to bomb:\n");

scanf\_s("%d%d", &row1, &col1);

}//While-end

system("cls");

}//While-End

if (count == 5)

{

printf("Congratulations! You have successfully bombed 5 ships!\n");

}//End-if

else

{

printf("Too bad! Better luck next time!\n");

}//End-if

for (x = 0; x < 10; x++)

{

for (y = 0; y < 10; y++)

{

map[x][y] = 0; //To reset the value in array to zero if the player wants to replay the game.

}//Loop-end

}//Loop-end

}//End of bomb function

|  |  |  |
| --- | --- | --- |
| **Input**  welcomeScreen  emptyMap  instructions  stage  bomb  level  start  bombrow  bombcol  noOfships  rows  col  x  y  coordinate  map[10][10]  col1  row1  count  attempt | **Processing**  🡪 Prints Game Title🡪  🡪 Prints Gameboard🡪  🡪 Prints Instructions🡪  🡪 Creates hidden ships by random coordinates🡪  🡪 Registers hidden ships for bombing🡪  🡪 Used to calculate wins for Player 1, Mr. Pacman🡪  🡪Request user input to start game🡪  🡪 Input for bomb row🡪  🡪 Input for bomb column🡪  🡪 sets number of ships to destroy 🡪  🡪 Number of columns in the gameboard🡪  🡪 Number of columns in the gameboard🡪  🡪 Input random x-axis location for hidden ship🡪  🡪 Input random y-axis location for hidden ship🡪  🡪 Used to specify where the bombs have been shot🡪  🡪 Creates gameboard by input🡪  🡪 Input y-axis coordinates for missle🡪  🡪 Input x-axis coordinates for missle🡪  🡪 Number of ships hit🡪  🡪 Number of attempts in destroying ships🡪 | **Output**  Prints Game Title  Prints Gameboard  Prints Instructions  Randomizes hidden ships  Bombs ships  Level 1 Game Play  Start Game or Quit  Output map[rows] Output map[col]  Number of ships to destroy  10 #’s  10 #’s  Output hidden ship location  Output hidden ship location  Identifies a miss or a hit  10 by 10 gameboard  Output where missle is shot by col and row  Output out of noOfships  Output number of attempts made |





