```c
#include <stdio.h>
#include <stdlib.h>
char board1[10][10]; //defines the two boards for the two players as global boards, that way they can be called any time without having to pass them through every function
char board2[10][10];
char board3[10][10];
char board4[10][10];
#define EMPTY 10 //defines all the states that the values of the arrays can be
#define MISS 16
#define HIT 17
#define FULL 18
#define STRIKE 19

void menu (){ //menu for starting the game, viewing rules, and quitting the application
int n;
printf("Battleship by Captain Nick Rodgers\n\n\n");
printf("Welcome! please select an option to continue\n");
printf("1: Start the game\n2: View rules \n3: Quit (this might delete system31) \n");
scanf ("%d",&n);
switch (n){
    case 1:
        reset();
        break;
    case 2:
        system("cls");
        rules();
        break;
    case 3:
        exit(0); //exits the application
        break;
}

}

void rules (void){ //prints the rules, when the user presses any key it returns to the menu
printf("Rules of Battleship:\n");
printf("1: Enter an X and Y coordinate, as well as a direction and a ship to place your ships\n");
printf("2: Players go in turns, don't screencheat like a little bitch\n");
printf("3: Guess coordinates of enemy ships by entering an X and Y coordinate into our highly advanced nuclear launch platform during your turn.\n Sink all ships to win!\n");
printf("(developers note: winner gets bragging rights and loser has to buy winner a beer)\n");
printf("\t\t\tPress any key to return to menu\n");
getch(); //scans for any key before procedding to the next command
system("cls"); //very handy operator, clears the screen making it hard to cheat and look at the other player's screen
menu();
}

void printboardplayer1 (){ //prints the board for player 1, all comments for this apply to printboardplayer2 and the printupperboards as well
system("cls"); //clears the screen to ensure no cheating and to keep things tidy.
printf("player 1 turn\n");
printupperboard1();
int i,j,n=0,m=0; //variables needed to traverse array as well as print the numbers along the sides of the board.
printf("  0 1 2 3 4 5 6 7 8 9\n"); // prints the top row of numbers: could be done with a loop but since battleship is a set size, this is simpler
printf("%d ",m);
for (i=0;i<10; i++){
    for (j=0; j<10 ;j++){ //traverses through the 2D array
            n=n+1; //increases counter that decides when row ends, when it reaches 10, a new row is created
        switch (board1[i][j]){ //simple switch case, to decide which symbol gets printed where in the board, enumeration makes this easy to read
    case EMPTY:
        printf ("~ "); //water tile
        break;
    case MISS:
        printf ("O "); //missed shot
        break;
    case FULL:
        printf ("I "); //location of part of a ship
        break;
    case STRIKE:
        printf ("X "); //where a ship has been hit
        break;
    default:
    break;

        }
    }
     m=m+1; //prints the number at the beginning of the row
     if (n==10){ //starts a new row
        printf("\n");
        n=0; //resets row counter
        if(m<10)
            printf("%d ",m);
     }
}
}

void printboardplayer2 (){ //prints the board for player 2
system("cls");
printf("player 2 turn\n");
printupperboard2();
int i,j,n=0,m=0;
printf("  0 1 2 3 4 5 6 7 8 9\n");
printf("%d ",m);
for (i=0;i<10; i++){
    for (j=0; j<10 ;j++){
            n=n+1;
        switch (board2[i][j]){
    case EMPTY:
        printf ("~ ");
        break;
    case MISS:
        printf ("O ");
        break;
    case FULL:
        printf ("I ");
        break;
    case STRIKE:
        printf ("X ");
        break;
    default:
    break;

        }
    }
     m=m+1;
     if (n==10){
        printf("\n");
        n=0;
        if(m<10)
            printf("%d ",m);
     }
}
}

void printupperboard1 (){ //prints upper board for player 1
int i,j,n=0,m=0;
printf("  0 1 2 3 4 5 6 7 8 9\n");
printf("%d ",m);
```

```c
for (i=0;i<10; i++){
    for (j=0; j<10 ;j++){
            n=n+1;
        switch (board3[i][j]){
    case EMPTY:
        printf ("~ ");
        break;
    case MISS:
        printf ("O ");
        break;
    case STRIKE:
        printf ("X ");
        break;
    default:
    break;

        }
    }
     m=m+1;
     if (n==10){
        printf("\n");
        n=0;
        if(m<10)
            printf("%d ",m);
     }
}
printf("_____________________\n");
}

void printupperboard2 (){ //prints upper board for player 1
int i,j,n=0,m=0;
printf("  0 1 2 3 4 5 6 7 8 9\n");
printf("%d ",m);
for (i=0;i<10; i++){
    for (j=0; j<10 ;j++){
            n=n+1;
        switch (board4[i][j]){
    case EMPTY:
        printf ("~ ");
        break;
    case MISS:
        printf ("O ");
        break;
    case STRIKE:
        printf ("X ");
        break;
    default:
    break;

        }
    }
     m=m+1;
     if (n==10){
        printf("\n");
        n=0;
        if(m<10)
            printf("%d ",m);
     }
}
printf("_____________________\n");
}

void player1taketurn(){ //allows player 1 to make their turn and prints their board, all comments apply to player2taketurn as well
printboardplayer1(); //prints the players upper and lower board
    int i,j; //variables for coordinates of attack
```

```c
printf("player 1, please enter coordinates to attack\n");
scanf("%d,%d",&i,&j); //scans the the user entered coordinates
switch (board2[i][j]){ //switch case to decide what to set the new value of the opposite board to at the chosen coordinate
case EMPTY:
    board2[i][j]=MISS;
    board3[i][j]=MISS;
    printf("missed!\n"); //if the coordinate on the opposite board is empty, a O will be printed there instead of the ~ water tile
    break;
case FULL:
    board2[i][j]=STRIKE;
    board3[i][j]=STRIKE;
    printf("hit!\n"); //if there is a ship at this coordinate in board 2, an X will be printed in place of the I ship to signify that it has been hit
    break;
}
printf("press any key to finish turn\n"); //waits for the player to be ready to end turn, to ensure no cheating
getch();
system("cls");
checkvictory(); //checks to see if either player has won so far
printf("player 2 turn\n");
player2taketurn(); //allows other player to go
}

void player2taketurn(){ //allows player 2 to make their turn and prints their board
printboardplayer2();
    int i,j;
printf("player 2, please enter coordinates to attack\n");
scanf("%d,%d",&i,&j);
switch (board1[i][j]){
case EMPTY:
    board1[i][j]=MISS;
    board4[i][j]=MISS;
    printf("missed!\n");
    break;
case FULL:
    board1[i][j]=STRIKE;
    board4[i][j]=STRIKE;
    printf("hit!\n");
    break;
}
printf("press any key to finish turn\n");
getch();
system("cls");
checkvictory();
player1taketurn(); //goes back to player 1, so that the pair can take turns until checkvictory finds a winner
}

void reset () { //initializes board to start a new game
    int i,j; //variables to traverse both arrays
    for (i=0;i<10; i++){
    for (j=0; j<10 ;j++){
        board1[i][j]=EMPTY;//resets both boards
        board2[i][j]=EMPTY;
        board3[i][j]=EMPTY;
        board4[i][j]=EMPTY;
    }
    }
system("cls");
printf ("board reset, ready to begin a new game\n");
setshipsplayer1();
}

```

```c
void setshipsplayer1(){ //gets coordinates, direction, and ship type from player 1 to set their ships, all comments apply to setshipsplayer2
system("cls");
printboardplayer1(); //prints board for player reference
int i,j,count,loop=4; //variables needed to traverse the array, and decide when the player cannot place any more ships
int c=1,b=1,d=1,p=1,n,direction; //a count for each ship to ensure that only one of every ship is printed to the board
printf("player 1, please place your ships\n");
printf("ships available: 4 long CARRIER, 3 long BATTLESHIP, 2 long DESTROYER, 1 long PATROL\n");
printf("please choose a ship (length), a coordinate set, and a direction (1 for horizontal, 2 for vertical) to place your ship\n");
while (loop!=0){ //loop variable gets smaller with each ship placed, allowing the player to place only 4 ships
printf("%d CARRIERs remaining, %d BATTLESHIPs remaining, %d DESTROYERs remaining, %d PATROLs remaining\n",c,b,d,p); //tells the player what ships they can still place
scanf("%d,%d,%d,%d",&n,&i,&j,&direction); //scans the ship to be placed, coordinate of head of ship, and direction to place the ship
if(n==1&&p==0){ //this set of if/else statements makes sure that the player cannot put down two of the same ship, by cross referencing the length they entered and the number of ships of that length remaining
    printf("no more PATROLs, try again\n");
    n=0;
    }
else if(n==2&&d==0){
    printf("no more DESTROYERs, try again\n");
    n=0;
    }
else if (n==3&&b==0){
    printf("no more BATTLESHIPs, try again\n");
    n=0;
    }
else if (n==4&&c==0){
    printf("no more CARRIERs, try again\n");
    n=0;
    }

if (direction==1){ //prints the ships according to direction that the player has chosen, 1 is horizontal, 2 is vertical
    for (count=0;count<n;count++)
        board1[i][j+count]=FULL;
}
else if (direction==2){
    for (count=0;count<n;count++)
        board1[i+count][j]=FULL;
}
if(n==1){ //sets the individual ship counts to zero once a ship has been placed, to ensure that only one of each ship can be placed
    p=0;
    loop=loop-1;
    }
else if(n==2){
    d=0;
    loop=loop-1;
    }
else if (n==3){
    b=0;
    loop=loop-1;
    }
else if (n==4){
    c=0;
    loop=loop-1;
    }
printboardplayer1(); //prints the board to show the current places of the ships
}
printf("press any key to let player 2 have a turn\n");
getch();
system("cls"); //clears screen to make ready for other player to enter their ships
setshipsplayer2();
}

void setshipsplayer2(){ //gets coordinates, direction, and ship type from player 2 to set their ships
system("cls");
printboardplayer2();
int i,j,count,loop=4;
int c=1,b=1,d=1,p=1,n,direction;
printf("player 2, please place your ships\n");
printf("ships available: 4 long CARRIER, 3 long BATTLESHIP, 2 long DESTROYER, 1 long PATROL\n");
printf("please choose a ship (length), a coordinate set, and a direction (1 for horizontal, 2 for vertical) to place your ship\n");
while (loop!=0){
printf("%d CARRIERs remaining, %d BATTLESHIPs remaining, %d DESTROYERs remaining, %d PATROLs remaining\n",c,b,d,p);
scanf("%d,%d,%d,%d",&n,&i,&j,&direction);
if(n==1&&p==0){
    printf("no more PATROLs, try again\n");
    n=0;
    }
else if(n==2&&d==0){
    printf("no more DESTROYERs, try again\n");
    n=0;
    }
else if (n==3&&b==0){
    printf("no more BATTLESHIPs, try again\n");
    n=0;
    }
else if (n==4&&c==0){
    printf("no more CARRIERs, try again\n");
    n=0;
    }

if (direction==1){
    for (count=0;count<n;count++)
        board2[i][j+count]=FULL;
}
else if (direction==2){
    for (count=0;count<n;count++)
        board2[i+count][j]=FULL;
}
if(n==1){
    p=0;
    loop=loop-1;
    }
else if(n==2){
    d=0;
    loop=loop-1;
    }
else if (n==3){
    b=0;
    loop=loop-1;
    }
else if (n==4){
    c=0;
    loop=loop-1;
    }
printboardplayer2();
}
printf("press any key to let player 1 have a turn\n");
getch();
system("cls");
player1taketurn();
}

```

```c
void checkvictory (){  //checks to see if a player has won yet and if so, declares the winner
int i,j; //variables to traverse both arrays
int p1=0, p2=0; //number of hits player 1 and 2 have
for (i=0;i<10;i++){
    for (j=0;j<10;j++){
        if (board3[i][j]==STRIKE) //goes through board 1 and counts up the total hits
            p1=p1+1;
        else if (board4[i][j]==STRIKE) //goes through board 2 and counts up the total hits
            p2=p2+1;
        else
            p1=p1;

        }
    }
if (p1==10){ // checks to see if player 1 has 10 hits, if so player 1 loses and player 2 wins
    printf ("\n\n\n\n\n+++++PLAYER 2 VICTORY!+++++\n\n\n\n\n");
    menu();
}
else if (p2==10){ // checks to see if player 1 has 10 hits, if so player 1 loses and player 1 wins
    printf ("\n\n\n\n\n=====PLAYER 1 VICTORY!=====\n\n\n\n\n");
    menu();
}
else
    return;

}

int main (){ //main program, calls the menu as soon as program starts
menu();
}
```