**Written**

**By**

**Muhammad Kaleem Ullah**

**Student Number: U2061122**

**Group 14**

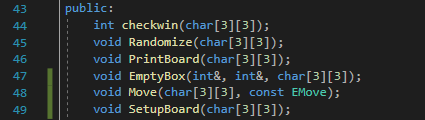
First of all I will describe the role of each functions and in the end I will discuss the flow program.

This program has 3 different files, and every file has a different important role.

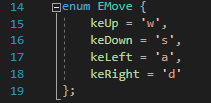
Hence the main function begins in **main.cpp** as given with Line number.



I will briefly discuss how this program works and how these functions conclude, and how their sequence works to the very end of this programs.



These functions are declared in **CW.h** which is known as the header file. Some of these functions are being called in other functions rather than in main functions.



Enum function in **CW.h** is used to set the set the character, which will be used to change the given location of empty spaces between the board.

First we are going to discuss the Setup board functions. This is declared on line 49 of **CW.h**

And the function is described in **CW.cpp** file.



void puzzle::SetupBoard(char caaBoard[3][3]) {

for (int iRow = 0; iRow < 3; ++iRow) {

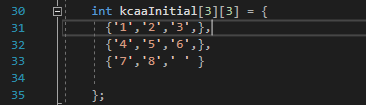
for (int iCol = 0; iCol < 3; ++iCol) {

caaBoard[iRow][iCol] = kcaaInitial[iRow][iCol];

}

}

}

The board is setup by a array described in CW.h file 

An empty array is given by char caaBoard[3][3]; in main.cpp and the for loop distribute the elements among the array. There are two loops form Row and Column. Then this function in directly called in main function,

After the Setup Board function the void PrintBoard(char[3][3]); After, the Print board function is used to display the given character on the board.

for (int iRow = 0; iRow < 3; iRow++) {

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

cout << " " << caaBoard[iRow][iCol];

cout << endl << endl;

}

it also uses 2x for loops 1 for rows and 1 for column, and then caaBoard array is used to store the given number from board to output on the console. So, a board like this is displayed



Now we are going to discuss the empty box functions this function is used to locate the free space, so that other numbers can be moved.

void puzzle::EmptyBox(int& irRow, int& irCol, char caaBoard[3][3]) {

for (int iRow = 0; iRow < 3; ++iRow) {

for (int iCol = 0; iCol < 3; ++iCol) {

if (caaBoard[iRow][iCol] == ' ') {

irRow = iRow;

irCol = iCol;

}

}

}

}

This is done by using if statement and 2 for loops, again loops are used for every space to be located.

These functions are used to randomize the element in arrays, this is only done single time in a given entry.

void puzzle::Randomize(char caaBoard[3][3])

srand((unsigned int)time(0));

for (int iIndex = 0; iIndex < 1000000; ++iIndex) {

const int kiNextMove = (rand() % 4);

switch (kiNextMove) {

case 0:

{

Move(caaBoard, keUp);

break;

}

case 1:

{

Move(caaBoard, keDown);

break;

}

case 2:

{

Move(caaBoard, keLeft);

break;

}

case 3:

{

Move(caaBoard, keRight);

break;

}

}

}

}

This functions work by srand and rand % the number of changes depend upon rand % lower the percentage lower the changes. So, this is used to control the flow of program.

The function checkwin() works by checking if the both the arrays possess the same characters.

int puzzle::checkwin(char caaBoard[3][3]) {

EmptyBox(iRowSpace, iColSpace, caaBoard);

int solved = 1;

for (int iRow = 0; iRow < 3; iRow++) {

for (int iCol = 0; iCol < 3; iCol++) {

if (caaBoard[iRow][iCol] != bcaaInitial[iRow][iCol])

solved = 0;

}

}

return solved;

}

All the other functions declared in this program are declared by using void this one is declared by int because this program has a return value, while void is a null which does not return a value. Hence this condition is how this works.

(caaBoard[iRow][iCol] != bcaaInitial[iRow][iCol])

Finally we are going to discuss the move functions, in this fuction there is some various aspects which are important. This functions also limits the blank space movements e.g. if blank is at corner and there is only 2 spaces only two movements are allowed, so the others do not work.

void puzzle::Move(char caaBoard[3][3], const EMove keMove) {

EmptyBox(iRowSpace, iColSpace, caaBoard);

int iRowMove(iRowSpace);

int iColMove(iColSpace);

switch (keMove) {

case keUp:

{

iRowMove = iRowSpace + 1;

break;

}

case keDown:

{

iRowMove = iRowSpace - 1;

break;

}

case keLeft:

{

iColMove = iColSpace + 1;

break;

}

case keRight:

{

iColMove = iColSpace - 1;

break;

}

}

if (iRowMove >= 0 && iRowMove < 3 && iColMove >= 0 && iColMove < 3) {

caaBoard[iRowSpace][iColSpace] = caaBoard[iRowMove][iColMove];

caaBoard[iRowMove][iColMove] = ' ';

}

}

The switch Is used to get the key Irow move is the array we are changing position in.

And finally in the If statement conditions restrict the movements of the black space as described earlier

First of all this program initialized the board element position. Then the randomize function is given so that whenever the program run it Randoms the board elements., then a do / while loop is used in the first of all the elements got printed on the console. And the input is taken from the user and variable cNextMove, which then send this to move function and program carries on. In the very last of do, the checkwin kicks in and checks if the program is done, hence checkwin runs with every key pressed, when both the array one which is used in program and the other which is used as constant just for comparison matches the program ends.