Project 2

Title: Mine-sweeping game

Course: CSC-5

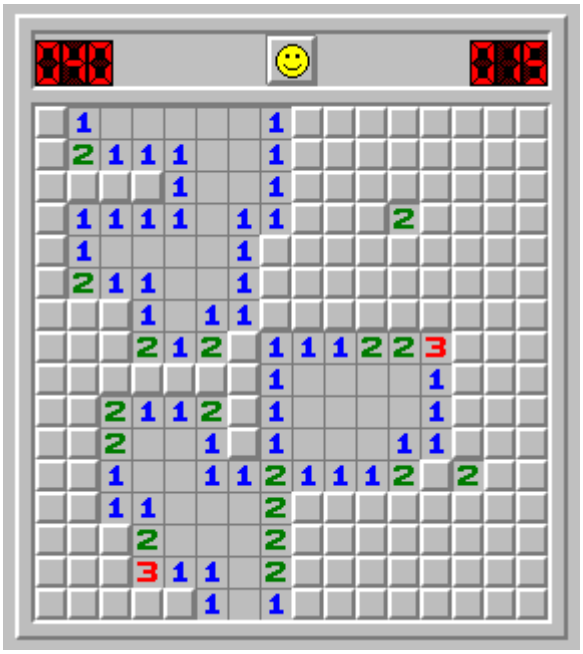
Section: 40717

Due Date: 2/12/2015

Author: Daniel Chang

Introduction:

Title: Mine-Sweeping



Minesweeper is a single-player puzzle video game. The objective of the game is to clear a rectangular board containing hidden "mines" without detonating any of them, with help from clues about the number of neighboring mines in each field. The player is initially presented with a grid of undifferentiated squares. Some randomly selected squares, unknown to the player, are designated to contain mines. The game is played by revealing squares of the grid by clicking or otherwise indicating each square. If a square containing a mine is revealed, the player loses the game. If no mine is revealed, a digit is instead displayed in the square, indicating how many adjacent squares contain mines; if no mines are adjacent, the square becomes blank. The player uses this information to deduce the contents of other squares, and may either safely reveal each square or mark the square as containing a mine.

Summary:

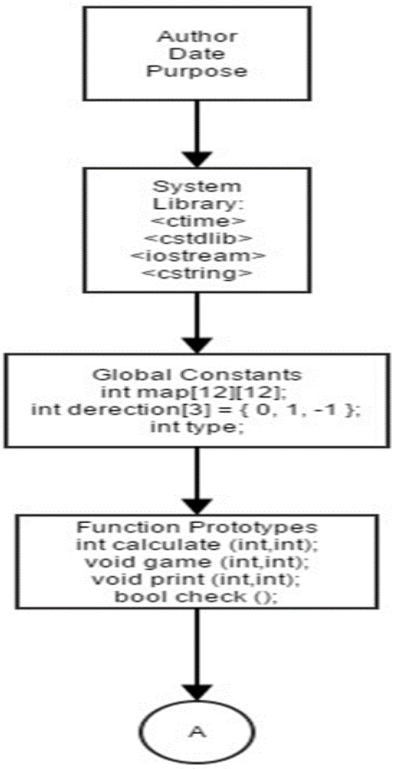
Project Size: 250+ Lines

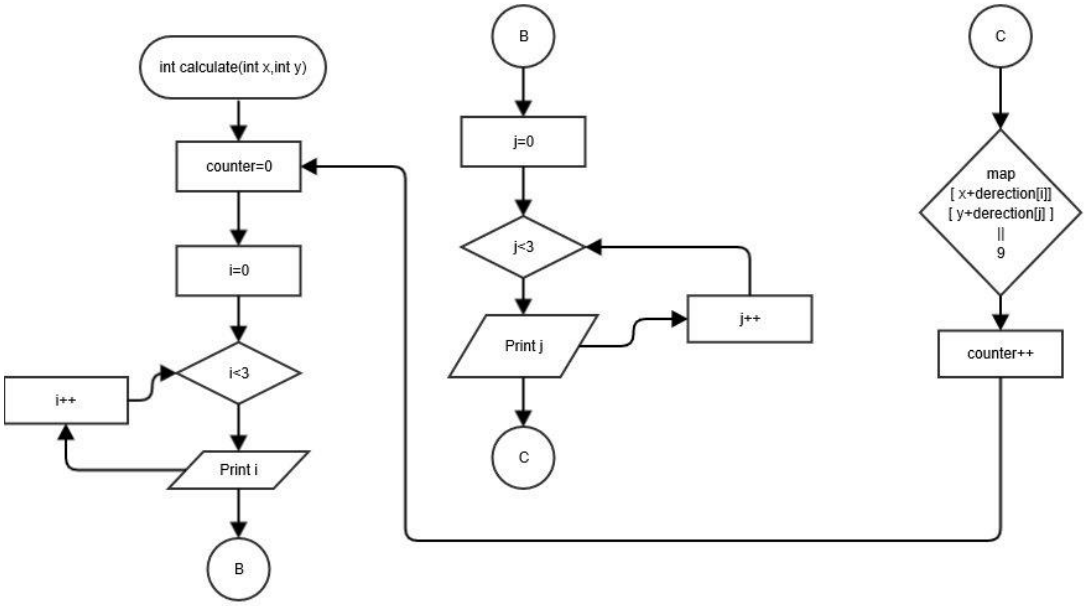
The number of Variables: 21

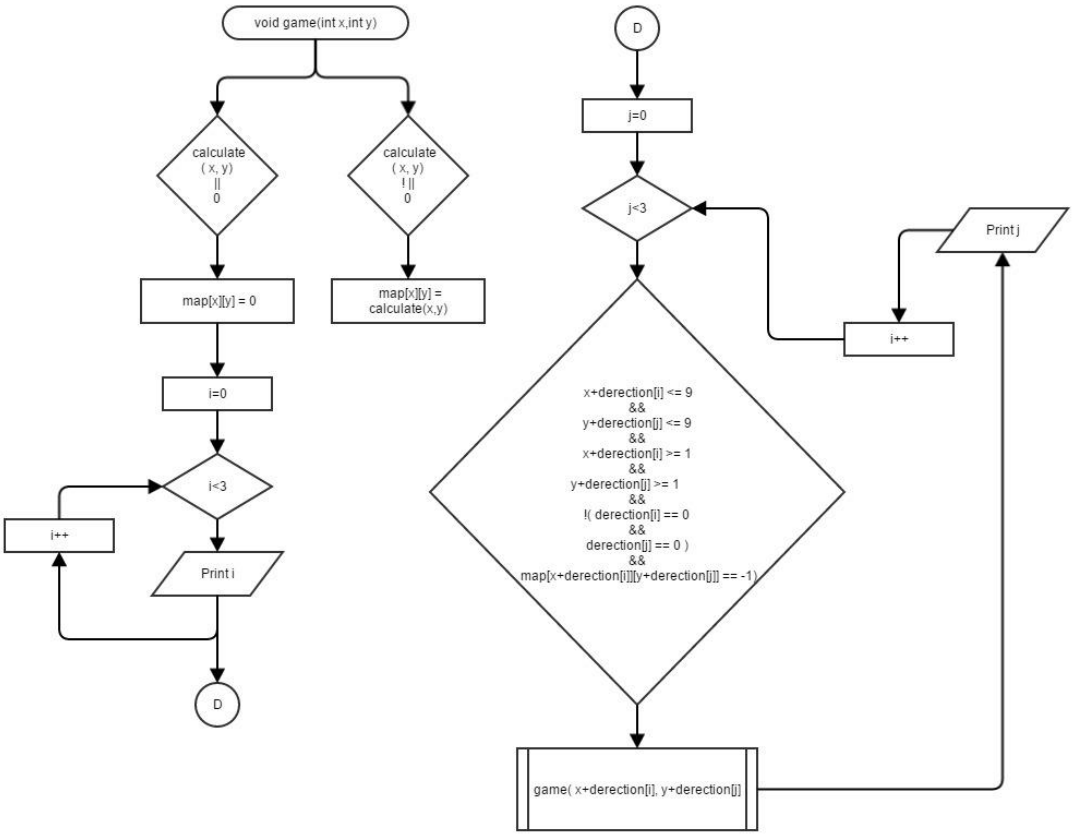
The number of function: 6

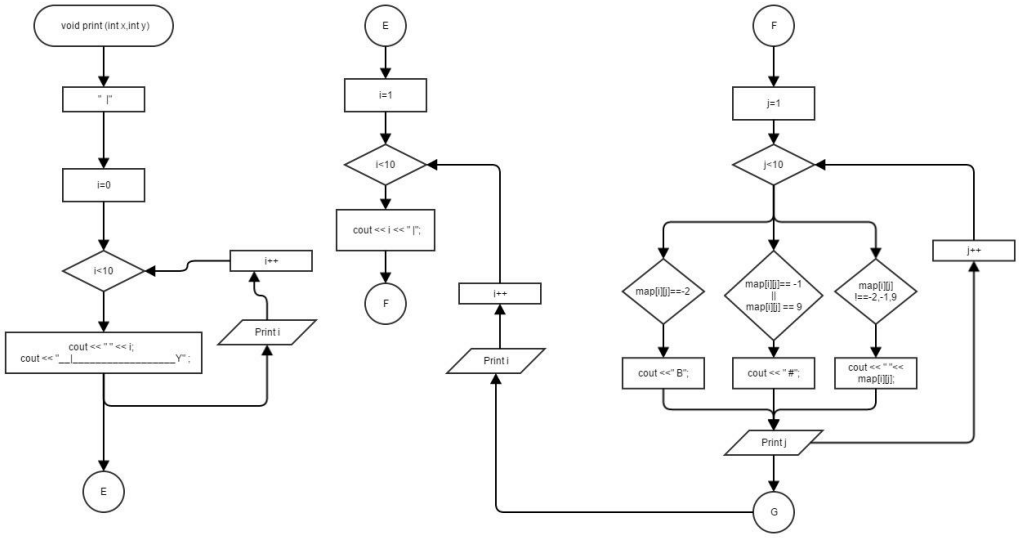
In this project, I try to use everything which I learned in the passing few weeks.
In the process I still have some method not understand, so I go through the textbooks and Google to review some proper noun. I programmed just a base game, and it still need to be completed. For example, one player game (with computer), and we only have to set up the probability of the computer. It took around one week. It was not very easy, most information are from textbooks, but I also have to search on the internet for getting more way to create the program. In the process to write the program, there have some variables that I learned from the google, and I used it on my program. I have a lot of problems, so I search on the web pages and textbooks and try to solved some of them. I know there still have another better way can do this project, but I just don’t know how to use it.

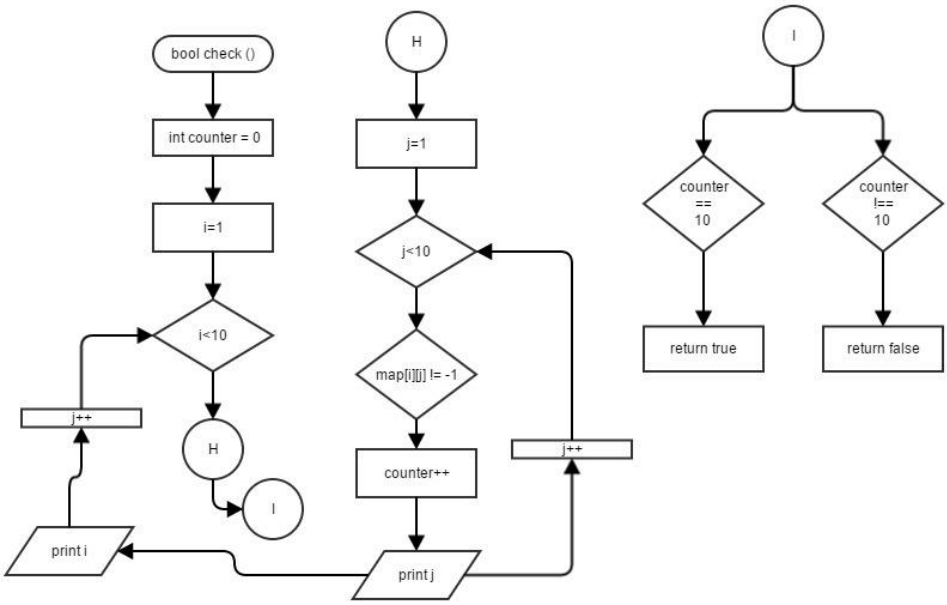
Flow-Chart:

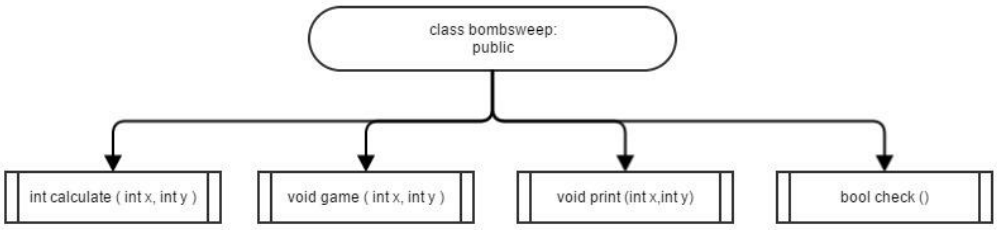


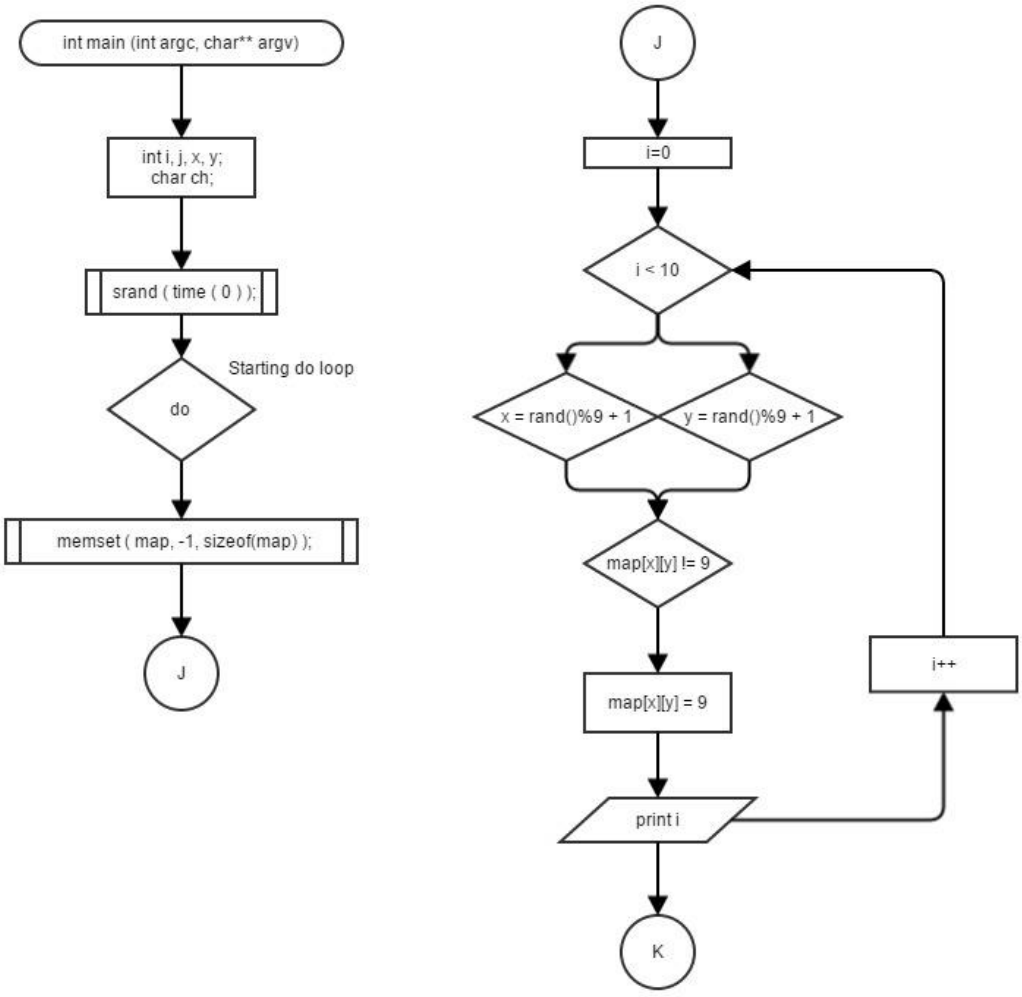

int calculate(int x,int y)
counter=0
i=0
i<3
i++
Print i
B
B
j=0
j<3
Print j
j++
C
C
map
[x+derection[i]]
[y+derection[j]]
||
9
counter++

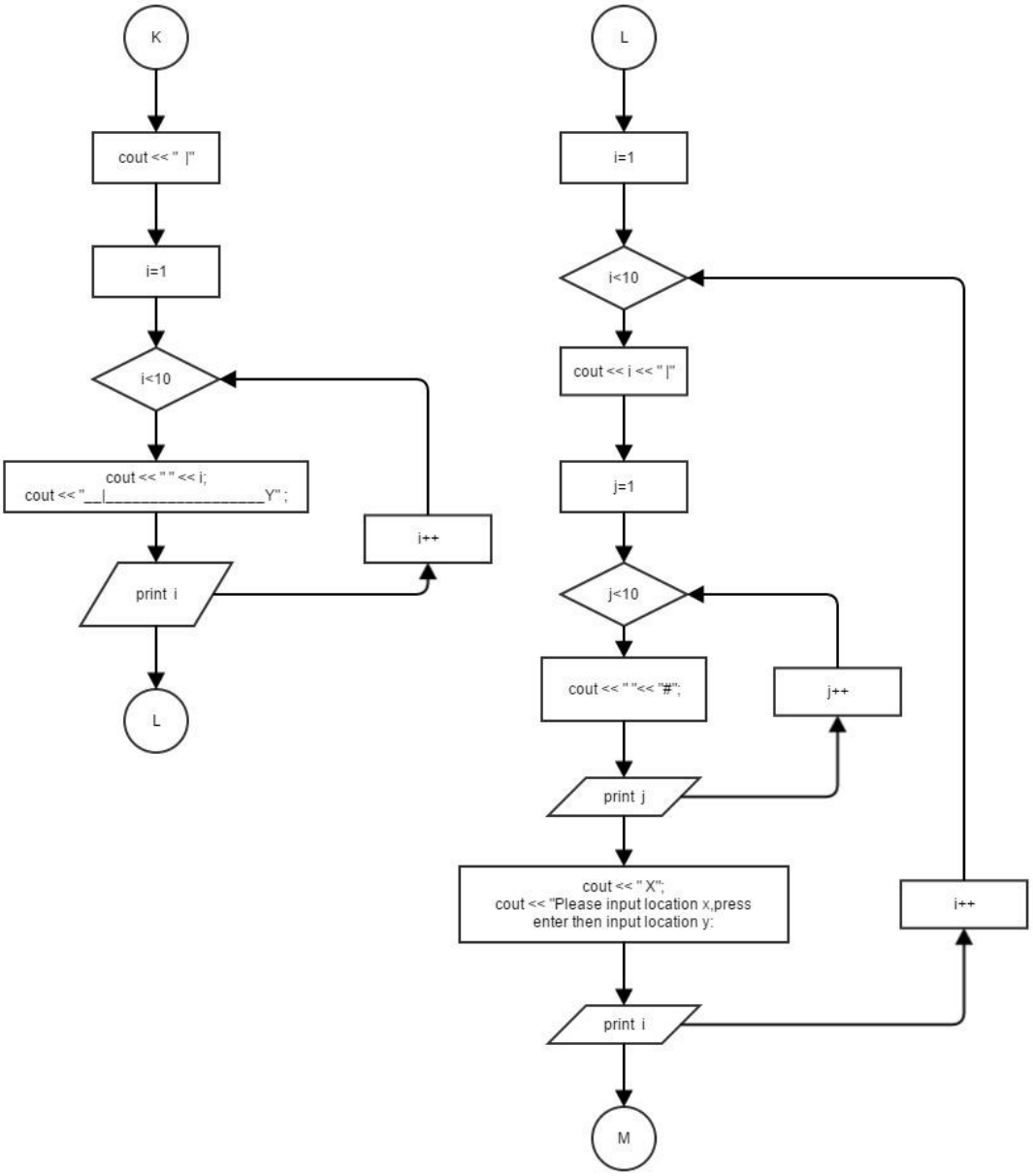

void game(int x,int y)
calculate
(x, y)
||
0
calculate
(x, y)
! ||
0
map[x][y] = 0
map[x][y] =
calculate(x,y)
i=0
i<3
i++
Print i
D
D
j=0
j<3
Print j
i++
x+derection[i] <= 9
&&
y+derection[j] <= 9
&&
x+derection[i] >= 1
&&
y+derection[j] >= 1
&&
!(derection[i] == 0
&&
derection[j] == 0)
&&
map[x+derection[i]][y+derection[j]] == -1)
game(x+derection[i], y+derection[j]

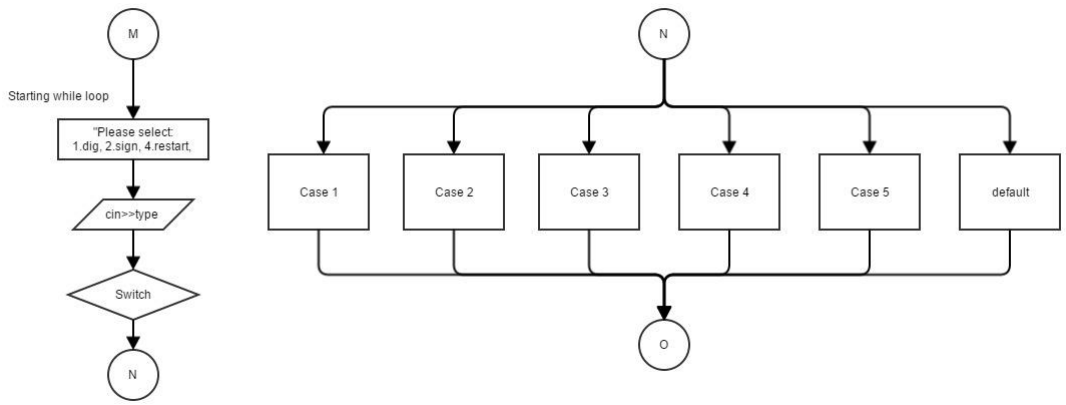

void print (int x,int y)
" |"
i=0
i<10
cout << " " << i;
cout << "__|_____Y" ;
i++
Print i
E
E
i=1
i<10
cout << i << " |";
F
i++
Print i
F
j=1
j<10
map[i][j]==-2
map[i][j]== -1
||
map[i][j] == 9
map[i][j]
!==-2,-1,9
cout <<" B";
cout << " #";
cout << " "<<
map[i][j];
j++
Print j
G

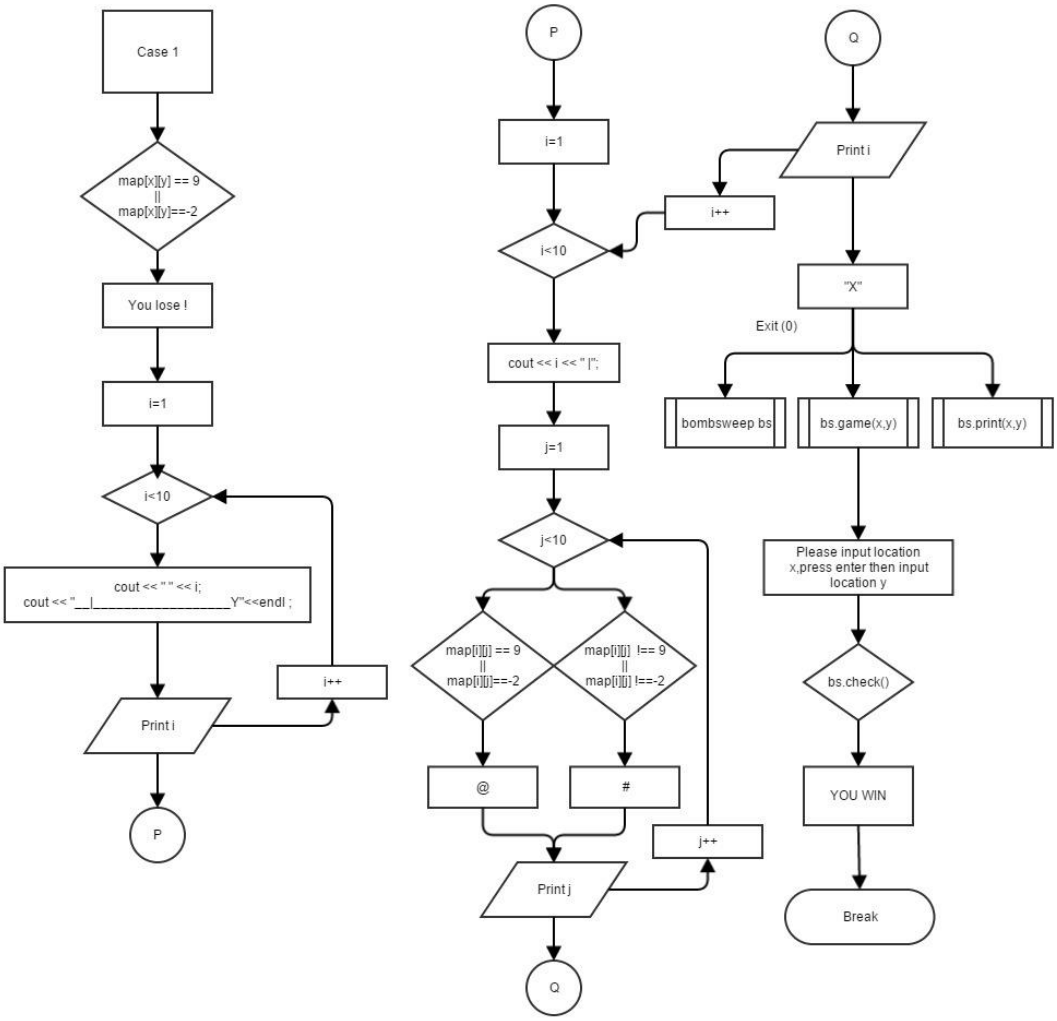

bool check ()
int counter = 0
i=1
i<10
j++
H
I
print i
H
j=1
j<10
map[i][j] != -1
counter++
print j
j++
I
counter
==
10
counter
!==
10
return true
return false

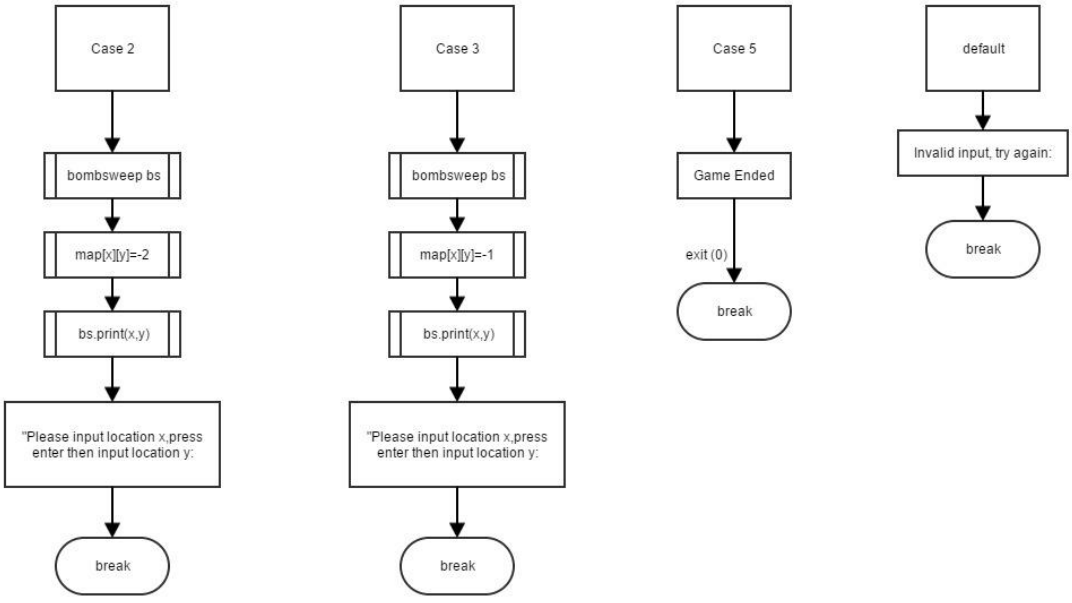

class bombsweep:
public
int calculate (int x, int y)
void game (int x, int y)
void print (int x,int y)
bool check ()

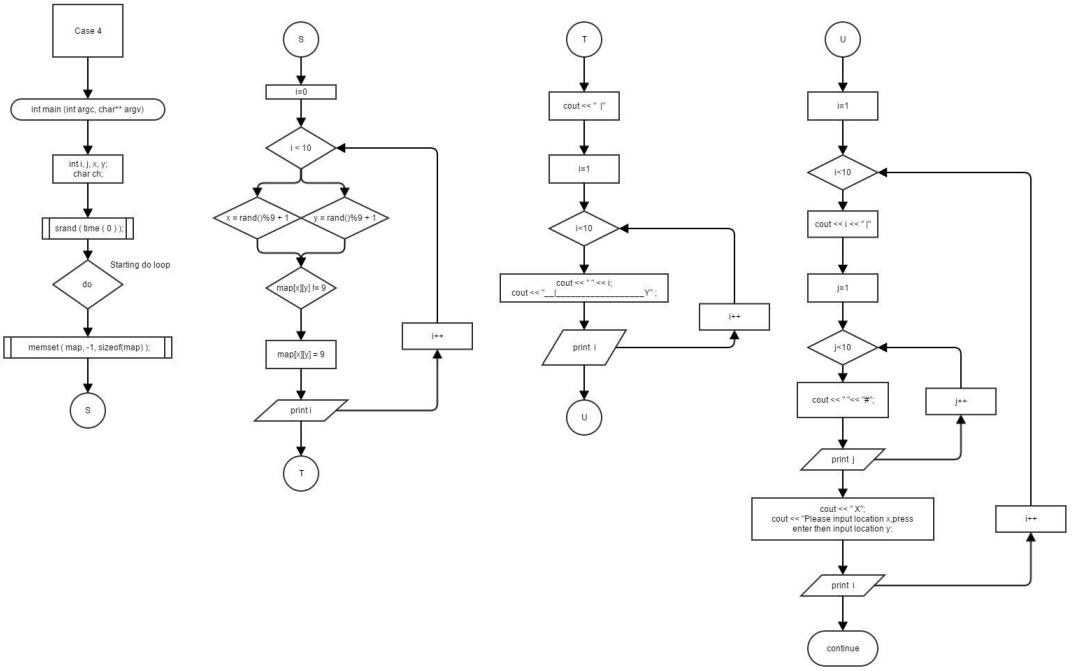

int main (int argc, char** argv)
int i, j, x, y;
char ch;
srand (time (0));
Starting do loop
do
memset (map, -1, sizeof(map));
J
J
i=0
i < 10
x = rand()%9 + 1
y = rand()%9 + 1
map[x][y] != 9
map[x][y] = 9
print i
i++
K

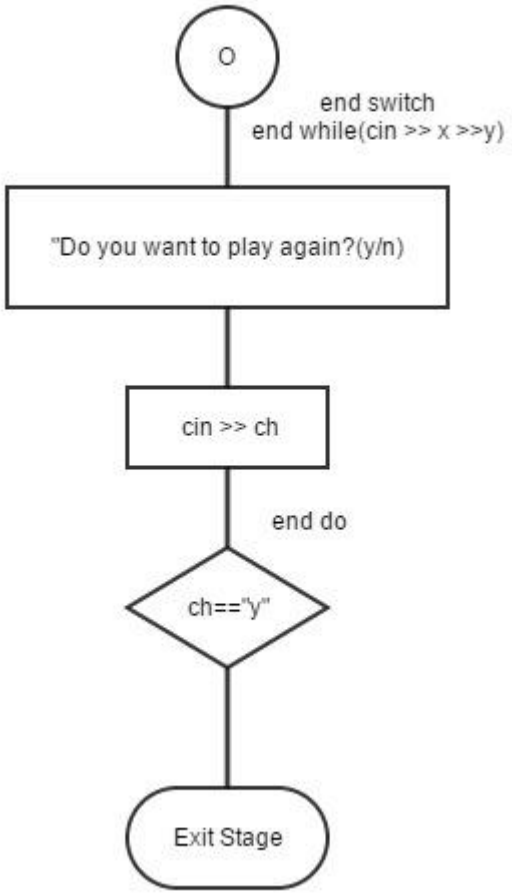

K
cout << " |"
i=1
i<10
cout << " " << i;
cout << "__|_____Y" ;
print i
i++
L
L
i=1
i<10
cout << i << " |"
j=1
j<10
cout << " "<< "#";
j++
print j
cout << " X";
cout << "Please input location x,press
enter then input location y:
print i
i++
M


M
Starting while loop
"Please select:
1.dig, 2.sign, 4.restart,
cin>>type
Switch
N
N
Case 1
Case 2
Case 3
Case 4
Case 5
default
O


Case 1
map[x][y] == 9
||
map[x][y]==-2
You lose !
i=1
i<10
cout << " " << i;
cout << "__|_____Y"<<endl ;
Print i
i++
P
P
i=1
i<10
cout << i << " |";
j=1
j<10
map[i][j] == 9
||
map[i][j]==-2
map[i][j] !== 9
||
map[i][j] !==-2
@
#
Print j
j++
Q
Q
Print i
i++
"X"
Exit (0)
bombsweep bs
bs.game(x,y)
bs.print(x,y)
Please input location x,press enter then input location y
bs.check()
YOU WIN
Break


Case 2
bombsweep bs
map[x][y]=-2
bs.print(x,y)
"Please input location x,press enter then input location y:
break
Case 3
bombsweep bs
map[x][y]=-1
bs.print(x,y)
"Please input location x,press enter then input location y:
break
Case 5
Game Ended
exit (0)
break
default
Invalid input, try again:
break


Case 4
int main (int argc, char** argv)
int i, j, x, y;
char ch;
srand (time (0));
Starting do loop
do
memset (map, -1, sizeof(map));
S
S
i=0
i < 10
x = rand()%9 + 1
y = rand()%9 + 1
map[x][y] != 9
map[x][y] = 9
print i
i++
T
T
cout << " |"
i=1
i<10
cout << " " << i;
cout << "__|_____________________Y";
print i
i++
U
U
i=1
i<10
cout << i << " |"
j=1
j<10
cout << " "<< "#";
print j
j++
cout << " X";
cout << "Please input location x,press
enter then input location y:
print i
i++
continue


O
end switch
end while(cin >> x >>y)
"Do you want to play again?(y/n)
cin >> ch
end do
ch=="y"
Exit Stage

Major Variables:

Variable Name:	**Description:**	**Location:**
x	x axis	int calculate (); void game (); void print (); bool check (); int main ()
y	Y axis	int calculate (); void game (); void print (); bool check (); int main ()
i	For the X' areas landmine	int calculate (); void game (); void print (); bool check (); int main ()
j	For the Y' areas landmine	int calculate (); void game (); void print (); bool check (); int main ()
ch	For repeat the game	int main ()
type	For dig, sign, and	int main ()

	restart	
map[ ][ ]	In order to avoid special treatment boundaries, it will expand the boundaries around a two-dimensional array	int calculate (); void game (); void print (); bool check (); int main ()
derection[ ]={ }	Array around the center point(There has 8 point)	int calculate (); void game ();
counter	Calculate the number of landmine around(x,y)'s 8 point	int calculate ();

C++ Constructs:

Key Words:	**Location:**
public class (class definition)	public class: bombsweep
If structure	if (calculate (x, y) == 0) {}
Equality operators and relational operators (==, !=, >, <, >=, <=)	x+derection[i] <= 9 && y+derection[j] derection[i] == 0 && derection[j] == 0

Arithmetic operators (+, -, *, /)	x+derection[i], y+derection[j]
Int primitive type	int i, j, x, y;
Void keyword	void game (int,int); void print (int,int);
If/else selection structure	if(map[i][j]==-2) cout <<" B"; else if (map[i][j] == -1 || map[i][j] == 9)

Increment operator (++)	if (map[x][y] != 9)
	map[x][y] = 9;
	i++;

While repetition structure	while (ch == 'y');
For repetition structure	for (i = 1; i < 10; i++) {}
Break	cout << "YOU WIN" << endl; break;

Return	if (counter == 10)
	return true;
	else
	return false;

Array	map[12][12]; derection[3] = { 0, 1, -1 };
Char primitive type	char ch;
Bool keyword	bool check ()
memset	memset (map, -1, sizeof(map));
Size of	memset (map, -1, sizeof(map));

Switch	cin >>type; switch(type) {}
Case	case 1: {}
exit	cout << " X\n"; exit(0);
continue	break; continue;

default	case 5:{
	cout << "Game Ended\n";
	exit(0);
	break;}
	default:{}

print	bs.print(x,y);

Reference:

http://pydoing.blogspot.com/2012/10/cpp-Loop.html

https://tw.knowledge.yahoo.com/question/question?qid=1510081702131

http://www.cplusplus.com/reference/cstring/memset/

http://linux.die.net/man/3/memset

http://www.cprogramming.com/tutorial/lesson8.html

http://www.puiching.edu.hk/~wtchung/trace/CPP/cpp13_c.htm

http://rs2.ocu.edu.tw/~jengchi/2Darray_usage.htm

http://www.openhome.cc/Gossip/JavaGossip-V1/ClassABC.htm

http://stackoverflow.com/questions/4792614/making-classes-public-to-other-classes-in-c

http://www.tutorialspoint.com/cplusplus/cpp_inheritance.htm

Program:

```
#include <ctime>
#include <cstdlib>
#include <iostream>
#include <cstring>

using namespace std;

int map[12][12];
int derection[3] = { 0, 1, -1 };
int type;

int calculate (int,int);
void game (int,int);
void print (int,int);
bool check ();


class bombsweep{

public:

      int calculate ( int x, int y ) {

            int counter = 0;
            for ( int i = 0; i < 3; i++ )
                  for ( int j = 0; j < 3; j++ )
                    if ( map[ x+derection[i]][ y+derection[j] ] == 9 )

                                          counter++;
                                    return counter;
```

```
}

void game ( int x, int y ) {

    if ( calculate ( x, y ) == 0 ) {

        map[x][y] = 0;
        for ( int i = 0; i < 3; i++ ) {



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

                    if ( x+derection[i] <= 9 && y+derection[j] <= 9 &&
x+derection[i] >= 1 && y+derection[j] >= 1
                            && !( derection[i] == 0 && derection[j] == 0 )
&&  map[x+derection[i]][y+derection[j]] == -1 )
                                    game( x+derection[i], y+derection[j] );
                }

            }else
                    map[x][y] = calculate(x,y);
              }


void print (int x,int y) {

    cout << "    |";
    for (int i=1; i<10; i++)
    cout << " " << i;
    cout << endl;
    cout << "__|_____________________Y" ;
    cout << endl;
    for ( int i = 1; i < 10; i++ ) {

        cout << i << " |";
    for ( int j = 1; j < 10; j++ ) {
```

```
                    if(map[i][j]==-2)
                        cout <<" B";
                    else if ( map[i][j] == -1 || map[i][j] == 9 )
                        cout << " #";
                    else
                        cout << " "<< map[i][j];
                }
                cout << "\n";
            }
            cout << "    X\n";
        }


        bool check () {

            int counter = 0;
            for ( int i = 1; i < 10; i++ ){
                for ( int j = 1; j < 10; j++ )
                    if ( map[i][j] != -1 )
                        counter++;
            }
            if ( counter == 10 )
                return true;
            else
                return false;
        }
};



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

        int i, j, x, y;
        char ch;
        srand ( time ( 0 ) );

        do {
```

```
memset ( map, -1, sizeof(map) );

for ( i = 0; i < 10;    ) {
        x = rand()%9 + 1;
        y = rand()%9 + 1;
        if ( map[x][y] != 9 ) {
                map[x][y] = 9;
                i++;
        }
}
cout << "    |";
for (i=1; i<10; i++)
        cout << " " << i;
cout << endl;
cout << "__|_____________________Y" ;
cout << endl;
for ( i = 1; i < 10; i++ ) {
        cout << i << " |";
        for ( j = 1; j < 10; j++ )
                cout << " "<< "#";
        cout << "\n";
}
cout << "    X\n";
cout << "Please input location x,press enter then input location y: \n";
while ( cin >> x >> y ) {
        cout << "Please select:1.dig, 2.sign, 4.restart,      \n";
        cin >>type;
        switch(type) {

        case 1: {

                        if ( map[x][y] == 9 || map[x][y]==-2) {
                        cout << "YOU LOSE!" << endl;
                        cout << "    |";
                        for (i=1; i<10; i++)
                                cout << " " << i;
                        cout << endl;
```

```
                cout << "__|____________________Y"<<endl ;
                for ( i = 1; i < 10; i++ ) {
                    cout << i << " |";
                    for ( j = 1; j < 10; j++ ) {
                        if ( map[i][j] == 9 || map[i][j]==-2)
                            cout << " @";
                        else
                            cout << " #";
                    }
                    cout << "\n";
                }
                cout << "    X\n";
                exit(0);
            }
            bombsweep bs;
            bs.game(x,y);
            bs.print(x,y);
            cout << "Please input location x,press enter then input location y:
\n";
            if ( bs.check()) {
                cout << "YOU WIN" << endl;
                break;
            }
            continue;
        }

        case 2: {
            bombsweep bs;
            map[x][y]=-2;
            bs.print(x,y);
            cout << "Please input location x,press enter then input location y:
\n";
            continue;
        }
        case 3: {
            bombsweep bs;
            map[x][y]=-1;
            bs.print(x,y);
```

```
                cout << "Please input location x,press enter then input location y:
\n";
                continue;
        }


        case 4:
        {
                memset ( map, -1, sizeof(map) );
                for ( i = 0; i < 10;    )
                {
                        x = rand()%9 + 1;
                        y = rand()%9 + 1;
                        if ( map[x][y] != 9 )
                       {
                                map[x][y] = 9;
                                i++;
                         }

                }

                cout << "    |";

                for (i=1; i<10; i++)
                        cout << " " << i;
                cout << endl;
                cout << "__|___________________Y" ;
                cout << endl;
                for ( i = 1; i < 10; i++ )

                {
                        cout << i << " |";

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

                                cout << " "<< "#";
                        cout << "\n";
                }
```

```
                cout << "    X\n";
                cout << "Please input location x,press enter then input location y:
\n";

                continue;

            }

            case 5:

                cout << "Game Ended\n";

                exit(0);

                break;

            default:

                cout<< "Invalid input, try again: \n";

                continue;

            }


        }

        cout << "Do you want to play again?(y/n):" << endl;

        cin >> ch;

    }

    while ( ch == 'y' );

    return 0;
}
```