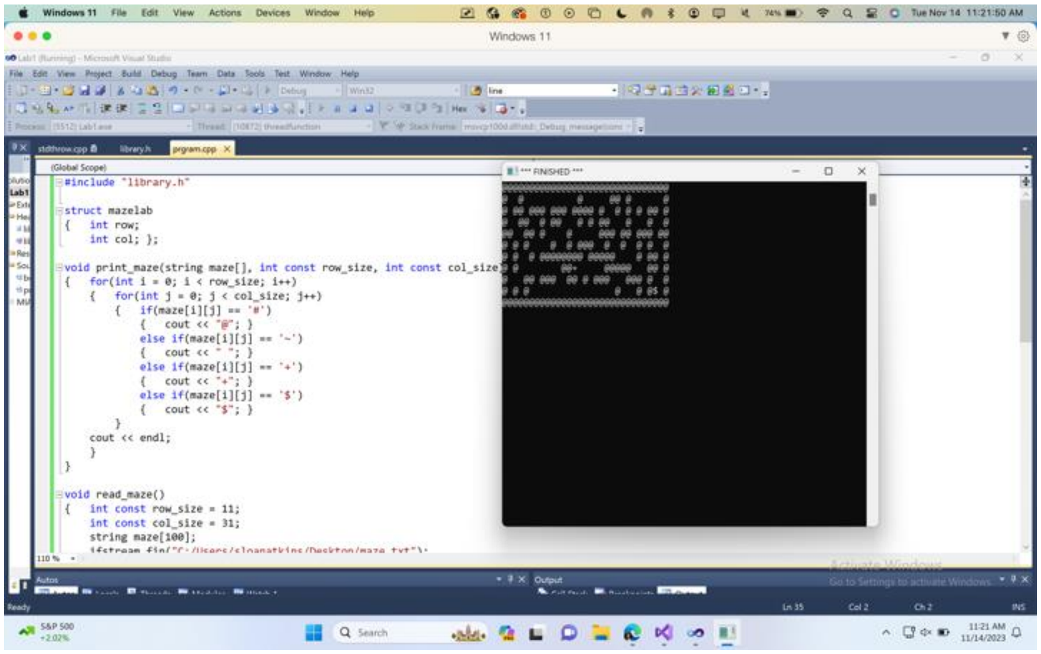
Lab 11

ECE 118 – Section R/RC
Lab on Wednesday at 5:05
Sloan Atkins

1. Read the Maze

```
#include "library.h"

struct mazelab
{      int row;
       int col; };

void print_maze(string maze[], int const row_size, int const col_size)
{      for(int i = 0; i < row_size; i++)
    {  for(int j = 0; j < col_size; j++)
        {    if(maze[i][j] == '#')
                   {      cout << "@"; }
             else if(maze[i][j] == '~')
                   {      cout << " "; }
             else if(maze[i][j] == '+')
                   {      cout << "+"; }
                   else if(maze[i][j] == '$')
                   {      cout << "$"; }
             }
    cout << endl;
       }
}

void read_maze()
{      int const row_size = 11;
       int const col_size = 31;
       string maze[100];
       ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
       if(fin.fail())
       {      cout << "Not available" << endl;}
              while(!fin.eof())
              {      for(int i = 0; i < 100; i++)
                     {      fin >> maze[i]; }
              fin.close();
              print_maze(maze, row_size, col_size); }}

void main()
{      read_maze(); }
```

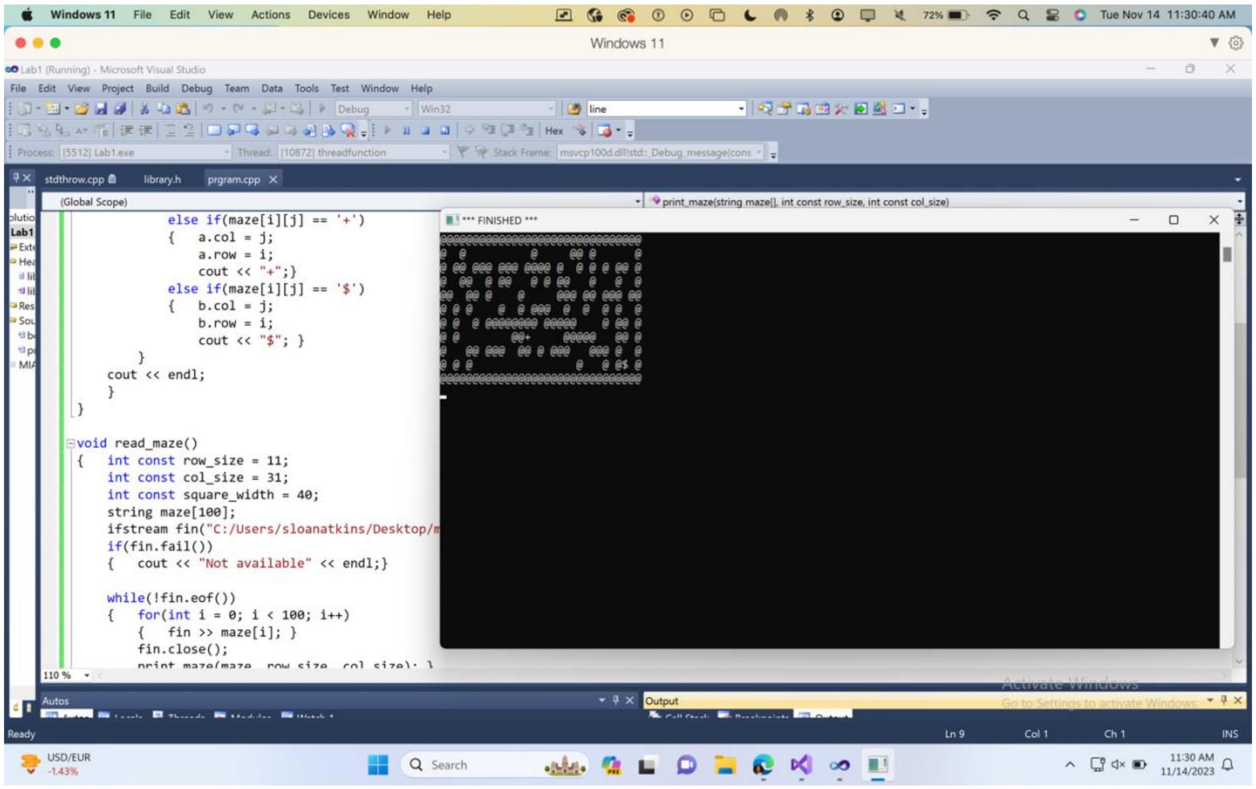

```
#include "library.h"

struct mazelab
{   int row;
    int col; };

void print_maze(string maze[], int const row_size, int const col_size)
{   for(int i = 0; i < row_size; i++)
    {   for(int j = 0; j < col_size; j++)
        {   if(maze[i][j] == '#')
            {   cout << "@"; }
            else if(maze[i][j] == '~')
            {   cout << " "; }
            else if(maze[i][j] == '+')
            {   cout << "+"; }
            else if(maze[i][j] == '$')
            {   cout << "$"; }
        }
    cout << endl;
    }
}

void read_maze()
{   int const row_size = 11;
    int const col_size = 31;
    string maze[100];
    ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
```

```
*** FINISHED ***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ @          @     @@ @       @
@ @@ @@@ @@@ @@@@ @  @ @ @ @@ @
@  @@  @ @@   @ @ @@   @   @  @
@@@  @@ @    @     @@@ @@ @@@ @@
@ @ @    @  @ @@@  @  @  @ @  @
@ @  @ @@@@@@@@ @@@@@    @ @@ @
@ @        @@+     @@@@@   @@ @
@  @@ @@@  @@ @ @@@    @@@ @  @
@ @ @                @   @ @$ @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

2. Detect + and $

```
#include "library.h"

struct mazelab
{       int row;
        int col; };

void print_maze(string maze[], int const row_size, int const col_size)
{       mazelab a, b;
        for(int i = 0; i < row_size; i++)
    {   for(int j = 0; j < col_size; j++)
          {     if(maze[i][j] == '#')
                        {       cout << "@"; }
                else if(maze[i][j] == '~')
                        {       cout << " "; }
                else if(maze[i][j] == '+')
                        {       a.col = j;
                        a.row = i;
                        cout << "+";}
                else if(maze[i][j] == '$')
                        {       b.col = j;
                                b.row = i;
                                cout << "$"; }
                }
        cout << endl;
        }
}

void read_maze()
{       int const row_size = 11;
```

```
        int const col_size = 31;
        int const square_width = 40;
        string maze[100];
        ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
        if(fin.fail())
        {       cout << "Not available" << endl;}

        while(!fin.eof())
        {       for(int i = 0; i < 100; i++)
                {       fin >> maze[i]; }
                fin.close();
                print_maze(maze, row_size, col_size); }
}

void main()
{       read_maze(); }
```



```
            else if(maze[i][j] == '+')
            {   a.col = j;
                a.row = i;
                cout << "+";}
            else if(maze[i][j] == '$')
            {   b.col = j;
                b.row = i;
                cout << "$"; }
        }
    cout << endl;
    }
}

void read_maze()
{   int const row_size = 11;
    int const col_size = 31;
    int const square_width = 40;
    string maze[100];
    ifstream fin("C:/Users/sloanatkins/Desktop/m
    if(fin.fail())
    {   cout << "Not available" << endl;}

    while(!fin.eof())
    {   for(int i = 0; i < 100; i++)
        {   fin >> maze[i]; }
        fin.close();
        print_maze(maze, row_size, col_size); }
```

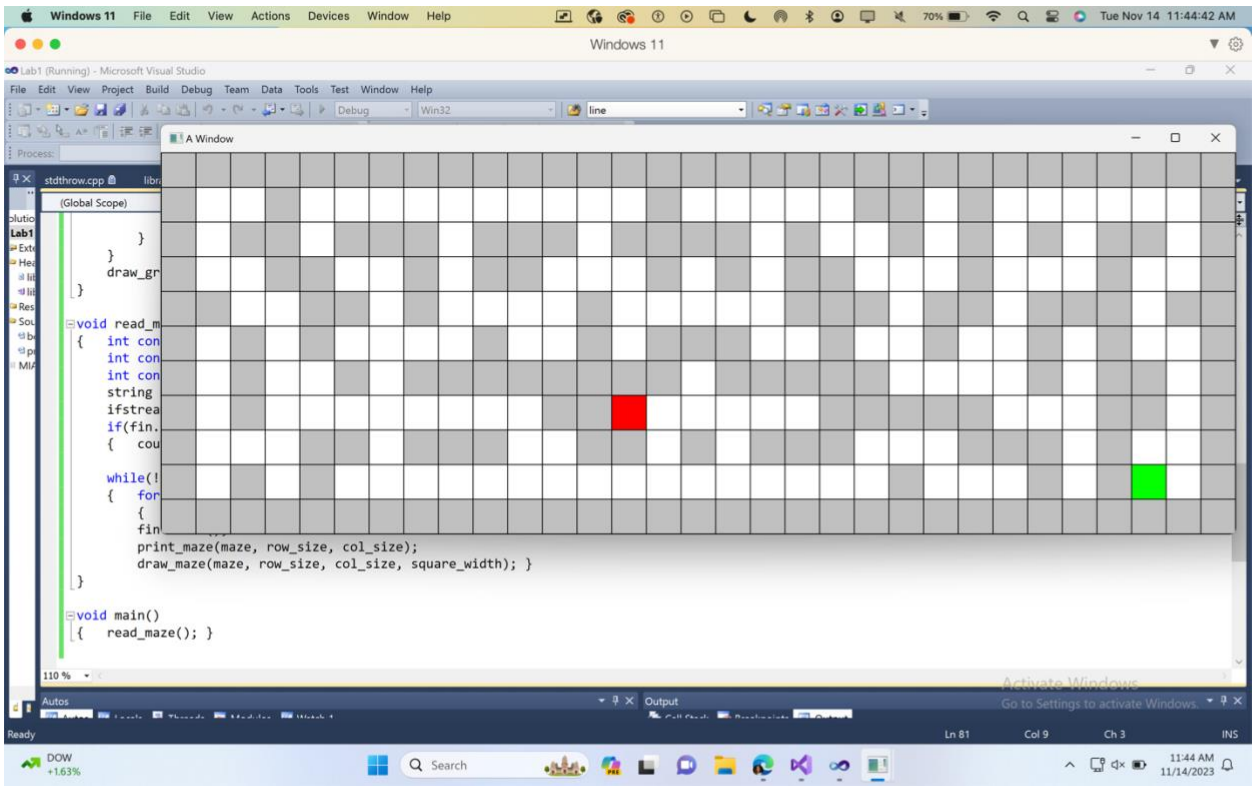
```
*** FINISHED ***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  @          @     @@ @      @
@ @@ @@@ @@@ @@@@ @  @ @ @ @@ @
@  @@  @ @@   @ @ @@   @   @  @
@@  @@ @    @     @@@ @@ @@@ @@
@ @ @    @  @ @@@  @  @  @ @  @
@ @  @ @@@@@@@@ @@@@@    @ @@ @
@ @        @@+     @@@@@   @@ @
@   @@ @@@  @@ @ @@@   @@@ @  @
@ @ @                @   @ @$ @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Windows 11
Lab1 (Running) - Microsoft Visual Studio
Process: [5512] Lab1.exe
Thread: [10872] threadfunction
Stack Frame: msvcp100d.dll!std::_Debug_message(cons
stdthrow.cpp
library.h
prgram.cpp
(Global Scope)
print_maze(string maze[], int const row_size, int const col_size)
110 %
Autos
Output
Ready
Ln 9
Col 1
Ch 1
INS

3. Draw it Properly

```
#include "library.h"

struct mazelab
{
int row;
int col; };
void print_maze(string maze[], int const row_size, int const col_size)
{       mazelab a, b;
        for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
                {       if(maze[i][j] == '#')
                        {       cout << "@"; }
                        else if(maze[i][j] == '~')
                        {       cout << " "; }
                        else if(maze[i][j] == '+')
                        {       a.col = j;
                                a.row = i;
                                cout << "+"; }
                        else if(maze[i][j] == '$')
                        {       b.col = j;
                                b.row = i;
                                cout << "$"; }
                }
                cout << endl;
        }
}
void draw_grid(int const row_size, int const col_size, int const square_width)
{       move_to(0,0);
        int r_width= row_size*square_width;
        int c_width= col_size*square_width;
        set_pen_width(1);
        set_pen_color(color::black);
        for(int i=0;i<row_size+1;i++)
        {       set_heading_degrees(90);
                draw_distance(c_width);
                move_relative(-c_width,square_width); }
        move_to(0,0);
        for(int i=0;i<col_size+1;i++)
        {       set_heading_degrees(180);
                draw_distance(r_width);
                move_relative(square_width,-r_width);
} }

void draw_maze(string maze[], int const row_size, int const col_size, int const
square_width)
{       int r_width= row_size*square_width;
        int c_width= col_size*square_width;
        make_window(c_width,r_width);
        for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
```

```
                {       if(maze[i][j]=='#')
                        {       set_pen_color(color::grey);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='~')
                        {       set_pen_color(color::white);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='+')
                        {       set_pen_color(color::red);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='$')
                        {       set_pen_color(color::green);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                }
        }
        draw_grid(row_size, col_size, square_width);
}

void read_maze()
{       int const row_size = 11;
        int const col_size = 31;
        int const square_width = 40;
        string maze[100];
        ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
        if(fin.fail())
        {       cout << "Not available" << endl;}

        while(!fin.eof())
        {       for(int i = 0; i < 100; i++)
                {       fin >> maze[i]; }
                fin.close();
                print_maze(maze, row_size, col_size);
                draw_maze(maze, row_size, col_size, square_width); }
}

void main()
{       read_maze(); }
```



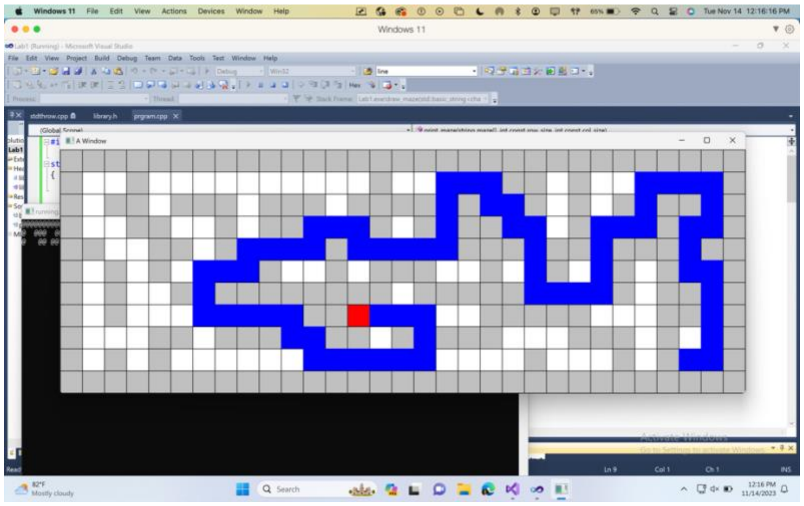
```
            }
        }
        draw_gr
}

void read_m
{   int con
    int con
    int con
    string
    ifstrea
    if(fin.
    {   cou

    while(!
    {   for
        {
        fin
        print_maze(maze, row_size, col_size);
        draw_maze(maze, row_size, col_size, square_width); }
}

void main()
{   read_maze(); }
```

4. Make the Robot Move

```
#include "library.h"

struct mazelab
{       int row;
        int col; };

void print_maze(string maze[], int const row_size, int const col_size)
{       for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
                {       if(maze[i][j] == '#')
                        {       cout << "@"; }
                        else if(maze[i][j] == '~')
                        {       cout << " "; }
                    else if(maze[i][j] == '+')
                        {       cout << "+"; }
                        else if(maze[i][j] == '$')
                        {       cout << "$"; }
                }
        }
}

void draw_grid(int const row_size, int const col_size, int const square_width)
{       move_to(0,0);
        int r_width= row_size*square_width;
```

```
	int c_width= col_size*square_width;
	set_pen_width(1);
	set_pen_color(color::black);
	for(int i=0;i<row_size+1;i++)
	{	set_heading_degrees(90);
		draw_distance(c_width);
		move_relative(-c_width,square_width); }
	move_to(0,0);
	for(int i=0;i<col_size+1;i++)
	{	set_heading_degrees(180);
		draw_distance(r_width);
		move_relative(square_width,-r_width); }
}

void move_robot(string maze[], mazelab a, mazelab b, int const square_width, int
const row_size, int const col_size)
{	int i = 0;
	double back_row[1000], back_col[1000];
	while(true)
	{	char c = wait_for_key_typed();
		if(c == -91)//LEFT
		{	back_row[i]=a.row;
			back_col[i]=a.col;
			a.col--;
			set_pen_color(color::blue);

	fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth); i++;
			cout << endl; }
		if(c == -89)//RIGHT
		{	back_row[i]=a.row;
			back_col[i]=a.col;
			a.col++;
			set_pen_color(color::blue);

	fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
			i++; }
		if(c == -90)//UP
		{	back_row[i]=a.row;
			back_col[i]=a.col;
			a.row--;
			set_pen_color(color::blue);

	fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
			i++; }
		if(c == -88)//DOWN
		{	back_row[i]=a.row;
			back_col[i]=a.col;
			a.row++;
			set_pen_color(color::blue);
	fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
			i++; }
		if(c == 'x')//EXIT
		{	break; }
	}
}

void draw_maze(string maze[], int const row_size, int const col_size, int const
square_width)
{	int r_width= row_size*square_width;
	int c_width= col_size*square_width;
	make_window(c_width,r_width);
	mazelab a, b;
	for(int i = 0; i < row_size; i++)
	{	for(int j = 0; j < col_size; j++)
		{	if(maze[i][j]=='#')
			{	set_pen_color(color::grey);

	fill_rectangle(j*square_width,i*square_width,square_width,square_width);}
			else if(maze[i][j]=='~')
			{	set_pen_color(color::white);

	fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
			else if(maze[i][j]=='+')
			{	a.col = j;
				a.row = i;
				set_pen_color(color::red);

	fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
			else if(maze[i][j]=='$')
			{	b.col = j;
				b.row = i;
				set_pen_color(color::green);

	fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
		}
	}
	draw_grid(row_size, col_size, square_width);
	move_robot(maze, a, b, square_width,row_size, col_size);
}

void read_maze()
{	int const row_size = 11;
	int const col_size = 31;
	int const square_width = 40;
	string maze[100];
	ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
	if(fin.fail())
	{	cout << "Not available" << endl; }
	while(!fin.eof())
```

```
        {       for(int i = 0; i < 100; i++)
                {       fin >> maze[i]; } }
        fin.close();
        print_maze(maze, row_size, col_size);
        draw_maze(maze, row_size, col_size, square_width);
}
void main()
{       read_maze(); }
```



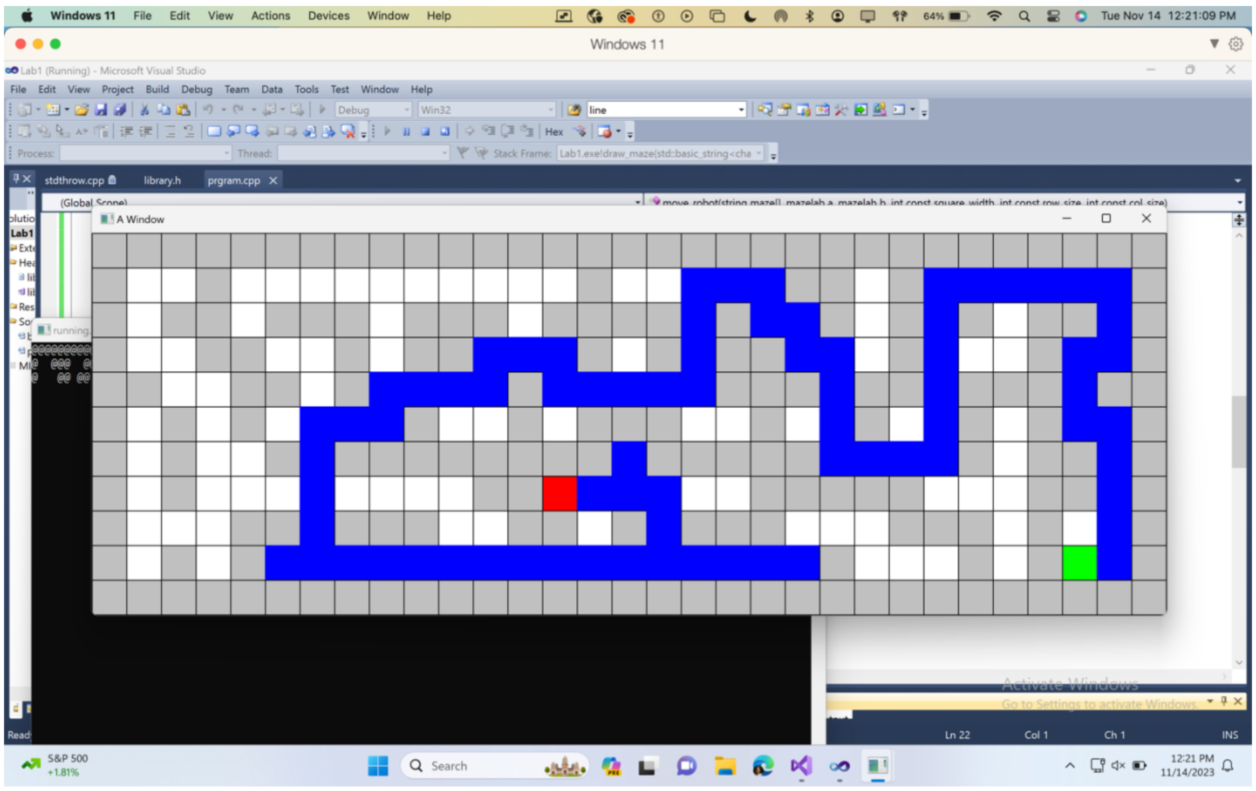
5. Prevent Walking Through Walls

```
#include "library.h"

struct mazelab
{       int row;
        int col; };

void print_maze(string maze[], int const row_size, int const col_size)
{       for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
                {       if(maze[i][j] == '#')
                        {       cout << "@"; }
                        else if(maze[i][j] == '~')
                        {       cout << " "; }
                    else if(maze[i][j] == '+')
                        {       cout << "+"; }
                        else if(maze[i][j] == '$')
                        {       cout << "$"; }
                }
        }
}
```

```
void draw_grid(int const row_size, int const col_size, int const square_width)
{      move_to(0,0);
       int r_width= row_size*square_width;
       int c_width= col_size*square_width;
       set_pen_width(1);
       set_pen_color(color::black);
       for(int i=0;i<row_size+1;i++)
       {      set_heading_degrees(90);
              draw_distance(c_width);
              move_relative(-c_width,square_width); }
       move_to(0,0);
       for(int i=0;i<col_size+1;i++)
       {      set_heading_degrees(180);
              draw_distance(r_width);
              move_relative(square_width,-r_width); }
}

void move_robot(string maze[], mazelab a, mazelab b, int const square_width, int
const row_size, int const col_size)
{      int i = 0;
       double back_row[1000], back_col[1000];
       while(true)
       {      char c = wait_for_key_typed();
              if(c == -91)
              {      if(maze[a.row][a.col-1] != '#')
             {      back_row[i]=a.row;
                            back_col[i]=a.col;
                            a.col--;
                            set_pen_color(color::blue);

       fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth); i++;
                            cout << endl; } }
              if(c == -89)
              {      if(maze[a.row][a.col+1] != '#')
             {      back_row[i]=a.row;
                            back_col[i]=a.col;
                            a.col++;
                            set_pen_color(color::blue);

       fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                            i++; } }
              if(c == -90)
              {      if(maze[a.row-1][a.col] != '#')
                     {      back_row[i]=a.row;
                            back_col[i]=a.col;
                            a.row--;
                            set_pen_color(color::blue);

       fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
```

```
                    i++; } }
            if(c == -88)
            {       if(maze[a.row+1][a.col] != '#')
                    {       back_row[i]=a.row;
                            back_col[i]=a.col;
                            a.row++;
                            set_pen_color(color::blue);

        fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                            i++; } }
            if(c == 'x')
            {       break; }
        }
}

void draw_maze(string maze[], int const row_size, int const col_size, int const
square_width)
{       int r_width= row_size*square_width;
        int c_width= col_size*square_width;
        make_window(c_width,r_width);
        mazelab a, b;
        for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
                {       if(maze[i][j]=='#')
                        {       set_pen_color(color::grey);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width);}
                        else if(maze[i][j]=='~')
                        {       set_pen_color(color::white);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='+')
                        {       a.col = j;
                                a.row = i;
                                set_pen_color(color::red);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='$')
                        {       b.col = j;
                                b.row = i;
                                set_pen_color(color::green);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                }
        }
        draw_grid(row_size, col_size, square_width);
        move_robot(maze, a, b, square_width,row_size, col_size);
}

void read_maze()
{       int const row_size = 11;
```

```
      int const col_size = 31;
      int const square_width = 40;
      string maze[100];
      ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
      if(fin.fail())
      {     cout << "Not available" << endl; }
      while(!fin.eof())
      {     for(int i = 0; i < 100; i++)
            {     fin >> maze[i]; } }
      fin.close();
      print_maze(maze, row_size, col_size);
      draw_maze(maze, row_size, col_size, square_width);
}
void main()
{     read_maze(); }
```



6. A Foolish Robot

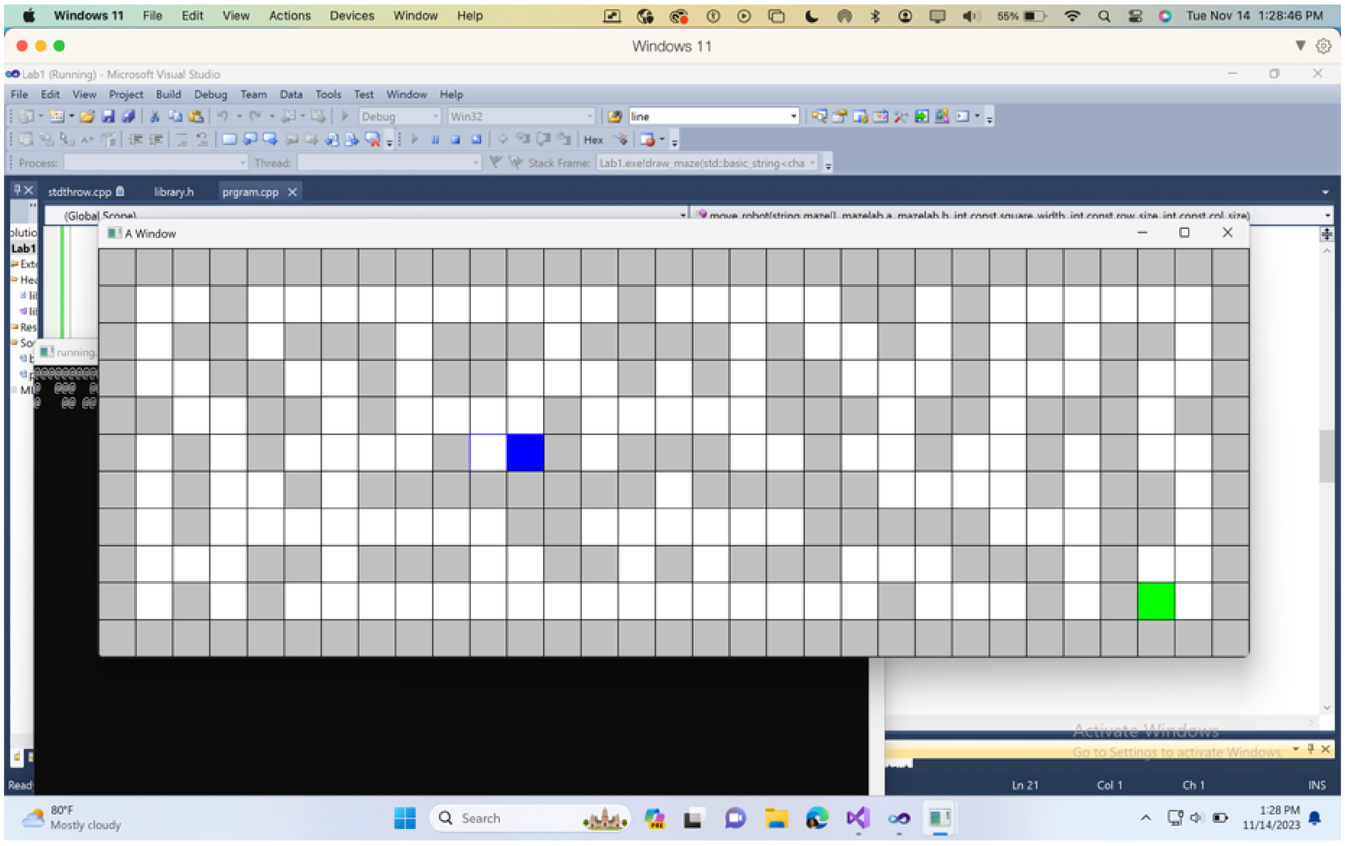
```
#include "library.h"

struct mazelab
{       int row;
        int col; };

void print_maze(string maze[], int const row_size, int const col_size)
{       for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
                {       if(maze[i][j] == '#')
                        {       cout << "@"; }
                        else if(maze[i][j] == '~')
                        {       cout << " "; }
                      else if(maze[i][j] == '+')
                        {       cout << "+"; }
                        else if(maze[i][j] == '$')
                        {       cout << "$"; }
                }
        }
}

void draw_grid(int const row_size, int const col_size, int const square_width)
{       move_to(0,0);
        int r_width= row_size*square_width;
        int c_width= col_size*square_width;
        set_pen_width(1);
        set_pen_color(color::black);
        for(int i=0;i<row_size+1;i++)
        {       set_heading_degrees(90);
                draw_distance(c_width);
                move_relative(-c_width,square_width); }
        move_to(0,0);
        for(int i=0;i<col_size+1;i++)
        {       set_heading_degrees(180);
                draw_distance(r_width);
                move_relative(square_width,-r_width); }
}

void move_robot(string maze[], mazelab a, mazelab b, int const square_width, int
const row_size, int const col_size)
{       int i = 0;
        double back_row[1000], back_col[1000];
        double been_there[11][31];
        while(true)
        {       char c = wait_for_key_typed();
                if(c == 'x')
                        exit(1);
                if(a.row == b.row && a.col == b.col)
                {       draw_grid(row_size, col_size, square_width);
                        for(int j = 0; back_row[j] >= 1 && back_row[j] <= 9; j++)
                        {       int row = back_row[j];
                                int col = back_col[j];
                                draw_grid(row_size, col_size, square_width); }
                        fill_rectangle(50,50,860,300,color::purple);
                        move_to(250,210);
                        set_font_size(80);
                        set_pen_color(color::yellow);
                        write_string("You have Won!!!");
                        break; }
                if(c == -91)
                {       if(maze[a.row][a.col-1] != '#')
                        {       set_pen_color(color::white);
                                fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                                draw_grid(row_size, col_size, square_width);
                                back_row[i] = a.row;
                                back_col[i] = a.col;
                                a.col--;
                                set_pen_color(color::blue);
                                fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                                i++; }
                }
                if(c == -89)
                {       if(maze[a.row][a.col+1] != '#')
                        {       set_pen_color(color::white);
                                fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                                draw_grid(row_size, col_size, square_width);
                                back_row[i]=a.row;
                                back_col[i]=a.col;
                                a.col++;
                                set_pen_color(color::blue);

        fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                                i++; }
                }
                if(c == -90)
                {       if(maze[a.row-1][a.col] != '#')
                        {       set_pen_color(color::white);
                                fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                                draw_grid(row_size, col_size, square_width);
                                back_row[i]=a.row;
                                back_col[i]=a.col;
                                a.row--;
                                set_pen_color(color::blue);

        fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                                i++; }
```

```
            }
            if(c == -88)
            {     if(maze[a.row+1][a.col] != '#')
                  {     set_pen_color(color::white);
                        fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                        draw_grid(row_size, col_size, square_width);
                        back_row[i]=a.row;
                        back_col[i]=a.col;
                        a.row++;
                        set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                        i++; }
                        }
            if(c == 'a')
            {     while(true)
                  {     char c = wait_for_key_typed(0.1);
                        back_row[i]=a.row;
                        back_col[i]=a.col;
                        if(a.row == b.row && a.col == b.col)
                        {     draw_grid(row_size, col_size, square_width);
                              for(int j = 0; back_row[j] >= 1 && back_row[j] <=
9; j++)

                              {     int row = back_row[j];
                                    int col = back_col[j];
                                    draw_grid(row_size, col_size,
square_width); }
                              fill_rectangle(50,50,860,300,color::purple);
                              move_to(250,210);
                              set_font_size(80);
                              set_pen_color(color::yellow);
                              write_string("You have Won!!!");
                              wait(2);
                              main(); }
                        if(c == 'm')
                        {     break;}
                        while(true)
                        {     if(maze[a.row][a.col - 1] != '#' &&
been_there[a.row][a.col - 1] != 1)
                              {     set_pen_color(color::white);
                                    fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
                                    draw_grid(row_size, col_size,
square_width);

                                    back_row[i] = a.row;
                                    back_col[i] = a.col;
                                    a.col--;
                                    set_pen_color(color::blue);
                                    fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
```

```
						i++;
						been_there[a.row][a.col] = 1;
						break; }
					else if(maze[a.row][a.col + 1] != '#' &&
been_there[a.row][a.col + 1] != 1)
					{	set_pen_color(color::white);
						fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
						draw_grid(row_size, col_size,
square_width);
						back_row[i]=a.row;
						back_col[i]=a.col;
						a.col++;
						set_pen_color(color::blue);

	fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
						i++;
						been_there[a.row][a.col] = 1;
						break; }
						else if(maze[a.row + 1][a.col] != '#' &&
been_there[a.row + 1][a.col] != 1)
					{	set_pen_color(color::white);
						fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
						draw_grid(row_size, col_size,
square_width);
						back_row[i]=a.row;
						back_col[i]=a.col;
						a.row++;
						set_pen_color(color::blue);

	fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
						i++;
						been_there[a.row][a.col] = 1;
						break; }
						else if(maze[a.row - 1][a.col] != '#' &&
been_there[a.row - 1][a.col] != 1)
					{	set_pen_color(color::white);
						fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
						draw_grid(row_size, col_size,
square_width);
						back_row[i]=a.row;
						back_col[i]=a.col;
						a.row--;
						set_pen_color(color::blue);

	fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
						i++;
```

```
                                      been_there[a.row][a.col] = 1;
                                      break; }
                                      else if(i > 0)
                                {     set_pen_color(color::white);
                                      fill_rectangle(a.col * square_width +1, 
a.row * square_width+1, square_width-1, square_width-1);
                                      i--;
                                      a.row = back_row[i];
                                      a.col = back_col[i];
                                      set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                                      break;}
                          }
                    }
            }
      }
}


void draw_maze(string maze[], int const row_size, int const col_size, int const 
square_width)
{     int r_width= row_size*square_width;
      int c_width= col_size*square_width;
      make_window(c_width,r_width);
      mazelab a, b;
      for(int i = 0; i < row_size; i++)
      {     for(int j = 0; j < col_size; j++)
            {     if(maze[i][j]=='#')
                  {     set_pen_color(color::grey);

      fill_rectangle(j*square_width,i*square_width,square_width,square_width);}
                  else if(maze[i][j]=='~')
                  {     set_pen_color(color::white);

      fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                  else if(maze[i][j]=='+')
                  {     a.col = j;
                        a.row = i;
                        set_pen_color(color::red);

      fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                  else if(maze[i][j]=='$')
                  {     b.col = j;
                        b.row = i;
                        set_pen_color(color::green);

      fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
            }
      }
      draw_grid(row_size, col_size, square_width);
```

```
        move_robot(maze, a, b, square_width,row_size, col_size);
}

void read_maze()
{       int const row_size = 11;
        int const col_size = 31;
        int const square_width = 40;
        string maze[100];
        ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
        if(fin.fail())
        {       cout << "Not available" << endl; }
        while(!fin.eof())
        {       for(int i = 0; i < 100; i++)
                {       fin >> maze[i]; } }
        fin.close();
        print_maze(maze, row_size, col_size);
        draw_maze(maze, row_size, col_size, square_width);
}
void main()
{       read_maze(); }
```


Windows 11
Lab1 (Running) - Microsoft Visual Studio
stdthrow.cpp
library.h
prgram.cpp
(Global Scope)
move_robot(string maze[], mazelab a, mazelab b, int const square_width, int const row_size, int const col_size)
A Window
Stack Frame: Lab1.exe!draw_maze(std::basic_string<cha
Ln 21
Col 1
Ch 1
INS
Tue Nov 14 1:28:46 PM
1:28 PM
11/14/2023
80°F
Mostly cloudy

7. Enemy

```
#include "library.h"

struct mazelab
{       int row;
        int col; };

void print_maze(string maze[], int const row_size, int const col_size)
{       for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
                {       if(maze[i][j] == '#')
                        {       cout << "@"; }
                        else if(maze[i][j] == '~')
                        {       cout << " "; }
                     else if(maze[i][j] == '+')
                        {       cout << "+"; }
                        else if(maze[i][j] == '$')
                        {       cout << "$"; }
                        else if(maze[i][j] == 'E')
                        {       cout << "E"; }
                }
        }
}

void draw_grid(int const row_size, int const col_size, int const square_width)
{       move_to(0,0);
        int r_width= row_size*square_width;
        int c_width= col_size*square_width;
        set_pen_width(1);
        set_pen_color(color::black);
        for(int i=0;i<row_size+1;i++)
        {       set_heading_degrees(90);
                draw_distance(c_width);
                move_relative(-c_width,square_width); }
        move_to(0,0);
        for(int i=0;i<col_size+1;i++)
        {       set_heading_degrees(180);
                draw_distance(r_width);
                move_relative(square_width,-r_width); }
}

void move_enemy(string maze[], mazelab m, mazelab b, int const square_width, int
const row_size, int const col_size)
{       int l = 0;
        double back_row[1000], back_col[1000];
        double been_there[11][31];
        while(true)
        {       char c = wait_for_key_typed();
                set_pen_color(color::green);
                fill_rectangle(b.col * square_width, b.row * square_width,
square_width, square_width);
```

```
            if(maze[m.row][m.col - 1] != '#' && been_there[m.row][m.col - 1] !=
1)
                    {       set_pen_color(color::white);
                            fill_rectangle(m.col * square_width, m.row *
square_width, square_width, square_width);
                            draw_grid(row_size, col_size, square_width);
                            back_row[l] = m.row;
                            back_col[l] = m.col;
                            m.col--;
                            set_pen_color(color::brown);
                            fill_rectangle(m.col * square_width, m.row *
square_width, square_width, square_width);
                            l++;
                            been_there[m.row][m.col] = 1; }
                    else if(maze[m.row][m.col + 1] != '#' &&
been_there[m.row][m.col + 1] != 1)//AUTO-RIGHT
                    {       set_pen_color(color::white);
                            fill_rectangle(m.col * square_width, m.row *
square_width, square_width, square_width);
                            draw_grid(row_size, col_size, square_width);
                            back_row[l]=m.row;
                            back_col[l]=m.col;
                            m.col++;
                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                            l++;
                            been_there[m.row][m.col] = 1; }
                    else if(maze[m.row + 1][m.col] != '#' && been_there[m.row +
1][m.col] != 1)
                    {       set_pen_color(color::white);
                            fill_rectangle(m.col * square_width, m.row *
square_width, square_width, square_width);
                            draw_grid(row_size, col_size, square_width);
                            back_row[l]=m.row;
                            back_col[l]=m.col;
                            m.row++;
                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                            l++;
                            been_there[m.row][m.col] = 1;}
                    else if(maze[m.row - 1][m.col] != '#' && been_there[m.row -
1][m.col] != 1)
                    {       set_pen_color(color::white);
                            fill_rectangle(m.col * square_width, m.row *
square_width, square_width, square_width);
                            draw_grid(row_size, col_size, square_width);
                            back_row[l]=m.row;
                            back_col[l]=m.col;
```

```
                                m.row--;
                                set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                l++;
                                been_there[m.row][m.col] = 1; }
                        else
                        {       set_pen_color(color::white);
                                fill_rectangle(m.col * square_width +1, m.row *
square_width+1, square_width-1, square_width-1);
                                l--;
                                m.row = back_row[l];
                                m.col = back_col[l];
                                set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth); } } }


void move_robot(string maze[], mazelab a, mazelab b, mazelab m, int const
square_width, int const row_size, int const col_size)
{       int i = 0;
        int l = 0;
        double back_row[1000], back_col[1000];
        double been_there[11][31];
        while(true)
        {       char c = wait_for_key_typed();
                set_pen_color(color::green);
                fill_rectangle(b.col * square_width, b.row * square_width,
square_width, square_width);
                if(c == 'x')
                        exit(1);
                if(a.row == b.row && a.col == b.col)
                {       draw_grid(row_size, col_size, square_width);
                        for(int j = 0; back_row[j] >= 1 && back_row[j] <= 9; j++)
                        {       int row = back_row[j];
                                int col = back_col[j];
                                draw_grid(row_size, col_size, square_width); }
                        fill_rectangle(50,50,860,300,color::purple);
                        move_to(250,210);
                        set_font_size(80);
                        set_pen_color(color::yellow);
                        write_string("You have Won!!!");
                        break; }
                if(m.row == a.row && m.col == a.col)
                {       draw_grid(row_size, col_size, square_width);
                        for(int j = 0; back_row[j] >= 1 && back_row[j] <= 9; j++)
                        {       int row = back_row[l];
                                int col = back_col[l];
                                draw_grid(row_size, col_size, square_width);}
                        fill_rectangle(50,50,860,300,color::white);
```

```
                  move_to(120,150);
                  set_font_size(80);
                  set_pen_color(color::dark_red);
                  write_string("You have been Caught!!!");
                  move_to(250,250);
                  write_string("GAME OVER");
                  break;}
            if(c == -91)
            {     if(maze[a.row][a.col-1] != '#')
                  {     set_pen_color(color::white);
                        fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                        draw_grid(row_size, col_size, square_width);
                        back_row[i] = a.row;
                        back_col[i] = a.col;
                        back_row[l] = m.row;
                        back_col[l] = m.col;
                        a.col--;
                        set_pen_color(color::blue);
                        fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                        i++;
                        {     if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
                              {     set_pen_color(color::white);
                                    fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                    draw_grid(row_size, col_size,
square_width);
                                    back_row[l] = m.row;
                                    back_col[l] = m.col;
                                    m.col--;
                                    set_pen_color(color::brown);
                                    fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                    l++;
                                    been_there[m.row][m.col] = 1; }
                              else if(maze[m.row][m.col + 1] != '#' &&
been_there[m.row][m.col + 1] != 1)
                              {     set_pen_color(color::white);
                                    fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                    draw_grid(row_size, col_size,
square_width);
                                    back_row[l]=m.row;
                                    back_col[l]=m.col;
                                    m.col++;
                                    set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                    l++;
```

```
                                        been_there[m.row][m.col] = 1; }
                                else if(maze[m.row + 1][m.col] != '#' &&
been_there[m.row + 1][m.col] != 1)
                                {       set_pen_color(color::white);
                                        fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                        draw_grid(row_size, col_size,
square_width);
                                        back_row[l]=m.row;
                                        back_col[l]=m.col;
                                        m.row++;
                                        set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                        l++;
                                        been_there[m.row][m.col] = 1;}
                                else if(maze[m.row - 1][m.col] != '#' &&
been_there[m.row - 1][m.col] != 1)
                                {       set_pen_color(color::white);
                                        fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                        draw_grid(row_size, col_size,
square_width);
                                        back_row[l]=m.row;
                                        back_col[l]=m.col;
                                        m.row--;
                                        set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                        l++;
                                        been_there[m.row][m.col] = 1; }
                                else
                                {       set_pen_color(color::white);
                                        fill_rectangle(m.col * square_width +1,
m.row * square_width+1, square_width-1, square_width-1);
                                        l--;
                                        m.row = back_row[l];
                                        m.col = back_col[l];
                                        set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth); } }

                        }
                }
                if(c == -89)
                {       if(maze[a.row][a.col+1] != '#')
                        {       set_pen_color(color::white);
                                fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
```

```
                         draw_grid(row_size, col_size, square_width);
                         back_row[i] = a.row;
                         back_col[i] = a.col;
                         back_row[l] = m.row;
                         back_col[l] = m.col;
                         a.col++;
                         set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                         i++;
                         {      if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
                                {      set_pen_color(color::white);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       draw_grid(row_size, col_size,
square_width);
                                       back_row[l] = m.row;
                                       back_col[l] = m.col;
                                       m.col--;
                                       set_pen_color(color::brown);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       l++;
                                       been_there[m.row][m.col] = 1; }
                                else if(maze[m.row][m.col + 1] != '#' &&
been_there[m.row][m.col + 1] != 1)
                                {      set_pen_color(color::white);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       draw_grid(row_size, col_size,
square_width);
                                       back_row[l]=m.row;
                                       back_col[l]=m.col;
                                       m.col++;
                                       set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                       l++;
                                       been_there[m.row][m.col] = 1; }
                                else if(maze[m.row + 1][m.col] != '#' &&
been_there[m.row + 1][m.col] != 1)
                                {      set_pen_color(color::white);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       draw_grid(row_size, col_size,
square_width);
                                       back_row[l]=m.row;
                                       back_col[l]=m.col;
                                       m.row++;
```

```
                                     set_pen_color(color::brown);

       fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                     l++;
                                     been_there[m.row][m.col] = 1;}
                              else if(maze[m.row - 1][m.col] != '#' &&
been_there[m.row - 1][m.col] != 1)
                              {      set_pen_color(color::white);
                                     fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                     draw_grid(row_size, col_size,
square_width);
                                     back_row[l]=m.row;
                                     back_col[l]=m.col;
                                     m.row--;
                                     set_pen_color(color::brown);

       fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                     l++;
                                     been_there[m.row][m.col] = 1; }
                              else
                              {      set_pen_color(color::white);
                                     fill_rectangle(m.col * square_width +1,
m.row * square_width+1, square_width-1, square_width-1);
                                     l--;
                                     m.row = back_row[l];
                                     m.col = back_col[l];
                                     set_pen_color(color::brown);

       fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth); } }
                      }
               }
               if(c == -90)
               {      if(maze[a.row-1][a.col] != '#')
                      {      set_pen_color(color::white);
                             fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                             draw_grid(row_size, col_size, square_width);
                             back_row[i]=a.row;
                             back_col[i]=a.col;
                             back_row[l] = m.row;
                             back_col[l] = m.col;
                             a.row--;
                             set_pen_color(color::blue);

       fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                             i++;
```

```
                         {      if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
                                {      set_pen_color(color::white);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       draw_grid(row_size, col_size,
square_width);
                                       back_row[l] = m.row;
                                       back_col[l] = m.col;
                                       m.col--;
                                       set_pen_color(color::brown);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       l++;
                                       been_there[m.row][m.col] = 1; }
                                else if(maze[m.row][m.col + 1] != '#' &&
been_there[m.row][m.col + 1] != 1)
                                {      set_pen_color(color::white);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       draw_grid(row_size, col_size,
square_width);
                                       back_row[l]=m.row;
                                       back_col[l]=m.col;
                                       m.col++;
                                       set_pen_color(color::brown);

    fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                       l++;
                                       been_there[m.row][m.col] = 1; }
                                else if(maze[m.row + 1][m.col] != '#' &&
been_there[m.row + 1][m.col] != 1)
                                {      set_pen_color(color::white);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                       draw_grid(row_size, col_size,
square_width);
                                       back_row[l]=m.row;
                                       back_col[l]=m.col;
                                       m.row++;
                                       set_pen_color(color::brown);

    fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                       l++;
                                       been_there[m.row][m.col] = 1;}
                                else if(maze[m.row - 1][m.col] != '#' &&
been_there[m.row - 1][m.col] != 1)
                                {      set_pen_color(color::white);
                                       fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
```

```
                                      draw_grid(row_size, col_size,
square_width);
                                      back_row[l]=m.row;
                                      back_col[l]=m.col;
                                      m.row--;
                                      set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                      l++;
                                      been_there[m.row][m.col] = 1; }
                                else
                                {     set_pen_color(color::white);
                                      fill_rectangle(m.col * square_width +1,
m.row * square_width+1, square_width-1, square_width-1);
                                      l--;
                                      m.row = back_row[l];
                                      m.col = back_col[l];
                                      set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth); } }}
            }
            if(c == -88)
            {     if(maze[a.row+1][a.col] != '#')
                  {     set_pen_color(color::white);
                        fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                        draw_grid(row_size, col_size, square_width);
                        back_row[i]=a.row;
                        back_col[i]=a.col;
                        back_row[l] = m.row;
                        back_col[l] = m.col;
                        a.row++;
                        set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                        i++;
                        {     if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
                              {     set_pen_color(color::white);
                                    fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                    draw_grid(row_size, col_size,
square_width);
                                    back_row[l] = m.row;
                                    back_col[l] = m.col;
                                    m.col--;
                                    set_pen_color(color::brown);
                                    fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
```

```
                                        l++;
                                        been_there[m.row][m.col] = 1; }
                                else if(maze[m.row][m.col + 1] != '#' &&
been_there[m.row][m.col + 1] != 1)
                                {       set_pen_color(color::white);
                                        fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                        draw_grid(row_size, col_size,
square_width);
                                        back_row[l]=m.row;
                                        back_col[l]=m.col;
                                        m.col++;
                                        set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                        l++;
                                        been_there[m.row][m.col] = 1; }
                                else if(maze[m.row + 1][m.col] != '#' &&
been_there[m.row + 1][m.col] != 1)
                                {       set_pen_color(color::white);
                                        fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                        draw_grid(row_size, col_size,
square_width);
                                        back_row[l]=m.row;
                                        back_col[l]=m.col;
                                        m.row++;
                                        set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                        l++;
                                        been_there[m.row][m.col] = 1;}
                                else if(maze[m.row - 1][m.col] != '#' &&
been_there[m.row - 1][m.col] != 1)
                                {       set_pen_color(color::white);
                                        fill_rectangle(m.col * square_width, m.row
* square_width, square_width, square_width);
                                        draw_grid(row_size, col_size,
square_width);
                                        back_row[l]=m.row;
                                        back_col[l]=m.col;
                                        m.row--;
                                        set_pen_color(color::brown);

        fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                        l++;
                                        been_there[m.row][m.col] = 1; }
                                else
                                {       set_pen_color(color::white);
```

```
                                        fill_rectangle(m.col * square_width +1, 
m.row * square_width+1, square_width-1, square_width-1);
                                        l--;
                                        m.row = back_row[l];
                                        m.col = back_col[l];
                                        set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth); } }
                                  }
                            }
            if(c == 'a')
            {     while(true)
                  {     set_pen_color(color::green);
                        fill_rectangle(b.col * square_width, b.row * 
square_width, square_width, square_width);
                        char c = wait_for_key_typed(0.1);
                        back_row[i]=a.row;
                        back_col[i]=a.col;
                        if(a.row == b.row && a.col == b.col)
                        {     draw_grid(row_size, col_size, square_width);
                              for(int j = 0; back_row[j] >= 1 && back_row[j] <= 
9; j++)
                              {     int row = back_row[j];
                                    int col = back_col[j];
                                    draw_grid(row_size, col_size, 
square_width); }
                              fill_rectangle(50,50,860,300,color::purple);
                              move_to(250,210);
                              set_font_size(80);
                              set_pen_color(color::yellow);
                              write_string("You have Won!!!");
                              wait(2);
                              main(); }
                        if(a.row == m.row && a.col == m.col)
                        {     draw_grid(row_size, col_size, square_width);
                              for(int j = 0; back_row[j] >= 1 && back_row[j] <= 
9; j++)
                              {     int row = back_row[j];
                                    int col = back_col[j];
                                    draw_grid(row_size, col_size, 
square_width); }
                              fill_rectangle(50,50,860,300,color::purple);
                              move_to(250,210);
                              set_font_size(80);
                              set_pen_color(color::yellow);
                              write_string("You have Lost!!!");
                              wait(2);
                              main(); }
                        if(c == 'm')
                        {     break;}
                        while(true)
```

```
                         {      if(maze[a.row][a.col - 1] != '#' &&
been_there[a.row][a.col - 1] != 1)
                               {      set_pen_color(color::white);
                                      fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
                                      draw_grid(row_size, col_size,
square_width);
                                      back_row[i] = a.row;
                                      back_col[i] = a.col;
                                      a.col--;
                                      set_pen_color(color::blue);
                                      fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
                                      i++;
                                      been_there[a.row][a.col] = 1;
                                      {      if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
                                             {      set_pen_color(color::white);
                                                    fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                    draw_grid(row_size, col_size,
square_width);
                                                    back_row[l] = m.row;
                                                    back_col[l] = m.col;
                                                    m.col--;
                                                    set_pen_color(color::brown);
                                                    fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                    l++;
                                                    been_there[m.row][m.col] = 1; }
                                             else if(maze[m.row][m.col + 1] != '#'
&& been_there[m.row][m.col + 1] != 1)
                                             {      set_pen_color(color::white);
                                                    fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                    draw_grid(row_size, col_size,
square_width);
                                                    back_row[l]=m.row;
                                                    back_col[l]=m.col;
                                                    m.col++;
                                                    set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                    l++;
                                                    been_there[m.row][m.col] = 1; }
                                             else if(maze[m.row + 1][m.col] != '#'
&& been_there[m.row + 1][m.col] != 1)
                                             {      set_pen_color(color::white);
                                                    fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
```

```
                                                        draw_grid(row_size, col_size, 
square_width);
                                                        back_row[l]=m.row;
                                                        back_col[l]=m.col;
                                                        m.row++;
                                                        set_pen_color(color::brown);
        
      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                        l++;
                                                        been_there[m.row][m.col] = 1;}
                                                  else if(maze[m.row - 1][m.col] != '#' 
&& been_there[m.row - 1][m.col] != 1)
                                                  {     set_pen_color(color::white);
                                                        fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                        draw_grid(row_size, col_size, 
square_width);
                                                        back_row[l]=m.row;
                                                        back_col[l]=m.col;
                                                        m.row--;
                                                        set_pen_color(color::brown);
        
      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                        l++;
                                                        been_there[m.row][m.col] = 1; }
                                                  else
                                                  {     set_pen_color(color::white);
                                                        fill_rectangle(m.col * 
square_width +1, m.row * square_width+1, square_width-1, square_width-1);
                                                        l--;
                                                        m.row = back_row[l];
                                                        m.col = back_col[l];
                                                        set_pen_color(color::brown);
        
      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                        been_there[m.row][m.col] = 1;}
}
                                            break; }
                                      else if(maze[a.row][a.col + 1] != '#' && 
been_there[a.row][a.col + 1] != 1)
                                      {     set_pen_color(color::white);
                                            fill_rectangle(a.col * square_width, a.row 
* square_width, square_width, square_width);
                                            draw_grid(row_size, col_size, 
square_width);
                                            back_row[i]=a.row;
                                            back_col[i]=a.col;
                                            back_row[l] = m.row;
                                            back_col[l] = m.col;
```

```
                                        a.col++;
                                        set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                                        i++;
                                        been_there[a.row][a.col] = 1;
                                        {      if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
                                               {      set_pen_color(color::white);
                                                      fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                      draw_grid(row_size, col_size,
square_width);
                                                      back_row[l] = m.row;
                                                      back_col[l] = m.col;
                                                      m.col--;
                                                      set_pen_color(color::brown);
                                                      fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                      l++;
                                                      been_there[m.row][m.col] = 1; }
                                               else if(maze[m.row][m.col + 1] != '#'
&& been_there[m.row][m.col + 1] != 1)
                                               {      set_pen_color(color::white);
                                                      fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                      draw_grid(row_size, col_size,
square_width);
                                                      back_row[l]=m.row;
                                                      back_col[l]=m.col;
                                                      m.col++;
                                                      set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                      l++;
                                                      been_there[m.row][m.col] = 1; }
                                               else if(maze[m.row + 1][m.col] != '#'
&& been_there[m.row + 1][m.col] != 1)
                                               {      set_pen_color(color::white);
                                                      fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                      draw_grid(row_size, col_size,
square_width);
                                                      back_row[l]=m.row;
                                                      back_col[l]=m.col;
                                                      m.row++;
                                                      set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
```

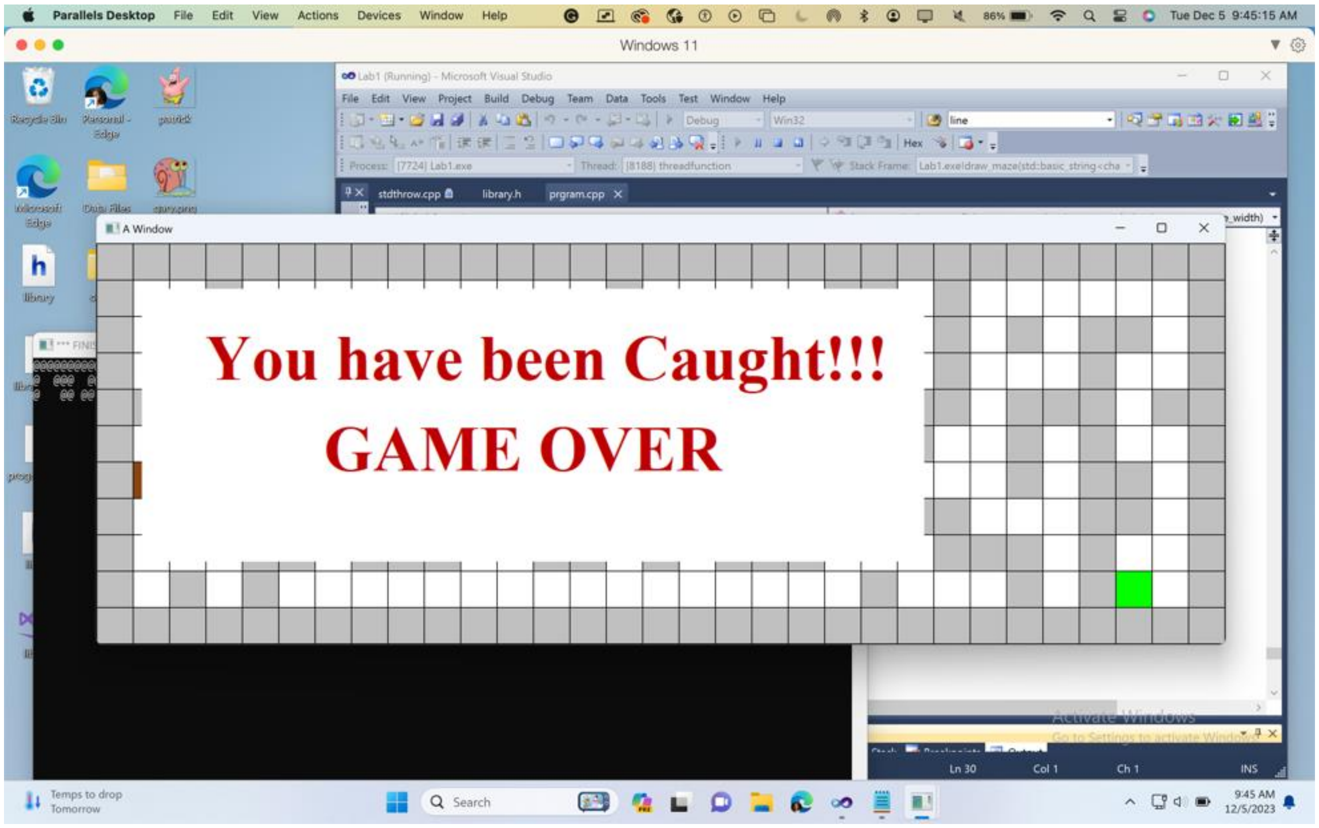
```
                                                        l++;
                                                        been_there[m.row][m.col] = 1;}
                                                  else if(maze[m.row - 1][m.col] != '#'
&& been_there[m.row - 1][m.col] != 1)
                                                  {     set_pen_color(color::white);
                                                        fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                        draw_grid(row_size, col_size,
square_width);

                                                        back_row[l]=m.row;
                                                        back_col[l]=m.col;
                                                        m.row--;
                                                        set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                        l++;
                                                        been_there[m.row][m.col] = 1; }
                                                  else
                                                  {     set_pen_color(color::white);
                                                        fill_rectangle(m.col *
square_width +1, m.row * square_width+1, square_width-1, square_width-1);
                                                        l--;
                                                        m.row = back_row[l];
                                                        m.col = back_col[l];
                                                        set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                        been_there[m.row][m.col] = 1;}
}
                                            break; }
                                            else if(maze[a.row + 1][a.col] != '#' &&
been_there[a.row + 1][a.col] != 1)
                                      {     set_pen_color(color::white);
                                            fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
                                            draw_grid(row_size, col_size,
square_width);

                                            back_row[i]=a.row;
                                            back_col[i]=a.col;
                                            back_row[l] = m.row;
                                            back_col[l] = m.col;
                                            a.row++;
                                            set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                                            i++;
                                            been_there[a.row][a.col] = 1;
                                            {     if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
```

```
                                            {      set_pen_color(color::white);
                                                   fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                   draw_grid(row_size, col_size, 
square_width);
                                                   back_row[l] = m.row;
                                                   back_col[l] = m.col;
                                                   m.col--;
                                                   set_pen_color(color::brown);
                                                   fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                   l++;
                                                   been_there[m.row][m.col] = 1; }
                                            else if(maze[m.row][m.col + 1] != '#' 
&& been_there[m.row][m.col + 1] != 1)
                                            {      set_pen_color(color::white);
                                                   fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                   draw_grid(row_size, col_size, 
square_width);
                                                   back_row[l]=m.row;
                                                   back_col[l]=m.col;
                                                   m.col++;
                                                   set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                   l++;
                                                   been_there[m.row][m.col] = 1; }
                                            else if(maze[m.row + 1][m.col] != '#' 
&& been_there[m.row + 1][m.col] != 1)
                                            {      set_pen_color(color::white);
                                                   fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                   draw_grid(row_size, col_size, 
square_width);
                                                   back_row[l]=m.row;
                                                   back_col[l]=m.col;
                                                   m.row++;
                                                   set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                   l++;
                                                   been_there[m.row][m.col] = 1;}
                                            else if(maze[m.row - 1][m.col] != '#' 
&& been_there[m.row - 1][m.col] != 1)
                                            {      set_pen_color(color::white);
                                                   fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                   draw_grid(row_size, col_size, 
square_width);
```

```
                                                        back_row[l]=m.row;
                                                        back_col[l]=m.col;
                                                        m.row--;
                                                        set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                        l++;
                                                        been_there[m.row][m.col] = 1; }
                                                  else
                                                  {     set_pen_color(color::white);
                                                        fill_rectangle(m.col *
square_width +1, m.row * square_width+1, square_width-1, square_width-1);
                                                        l--;
                                                        m.row = back_row[l];
                                                        m.col = back_col[l];
                                                        set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                        been_there[m.row][m.col] = 1;}
}
                                            break; }
                                            else if(maze[a.row - 1][a.col] != '#' &&
been_there[a.row - 1][a.col] != 1)
                                      {     set_pen_color(color::white);
                                            fill_rectangle(a.col * square_width, a.row
* square_width, square_width, square_width);
                                            draw_grid(row_size, col_size,
square_width);
                                            back_row[i]=a.row;
                                            back_col[i]=a.col;
                                            back_row[l] = m.row;
                                            back_col[l] = m.col;
                                            a.row--;
                                            set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                                            i++;
                                            been_there[a.row][a.col] = 1;
                                            {     if(maze[m.row][m.col - 1] != '#' &&
been_there[m.row][m.col - 1] != 1)
                                                  {     set_pen_color(color::white);
                                                        fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                        draw_grid(row_size, col_size,
square_width);
                                                        back_row[l] = m.row;
                                                        back_col[l] = m.col;
                                                        m.col--;
                                                        set_pen_color(color::brown);
```

```
                                                            fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                            l++;
                                                            been_there[m.row][m.col] = 1; }
                                                    else if(maze[m.row][m.col + 1] != '#' 
&& been_there[m.row][m.col + 1] != 1)
                                                    {       set_pen_color(color::white);
                                                            fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                            draw_grid(row_size, col_size, 
square_width);
                                                            back_row[l]=m.row;
                                                            back_col[l]=m.col;
                                                            m.col++;
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            l++;
                                                            been_there[m.row][m.col] = 1; }
                                                    else if(maze[m.row + 1][m.col] != '#' 
&& been_there[m.row + 1][m.col] != 1)
                                                    {       set_pen_color(color::white);
                                                            fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                            draw_grid(row_size, col_size, 
square_width);
                                                            back_row[l]=m.row;
                                                            back_col[l]=m.col;
                                                            m.row++;
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            l++;
                                                            been_there[m.row][m.col] = 1;}
                                                    else if(maze[m.row - 1][m.col] != '#' 
&& been_there[m.row - 1][m.col] != 1)
                                                    {       set_pen_color(color::white);
                                                            fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                            draw_grid(row_size, col_size, 
square_width);
                                                            back_row[l]=m.row;
                                                            back_col[l]=m.col;
                                                            m.row--;
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            l++;
                                                            been_there[m.row][m.col] = 1; }
                                                    else
                                                    {       set_pen_color(color::white);
                                                            fill_rectangle(m.col * 
square_width +1, m.row * square_width+1, square_width-1, square_width-1);
                                                            l--;
                                                            m.row = back_row[l];
                                                            m.col = back_col[l];
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            been_there[m.row][m.col] = 1;}
}
                                                    break; }
                                                    else if(i > 0)
                                            {       set_pen_color(color::white);
                                                    fill_rectangle(a.col * square_width +1, 
a.row * square_width+1, square_width-1, square_width-1);
                                                    i--;
                                                    a.row = back_row[i];
                                                    a.col = back_col[i];
                                                    back_row[l] = m.row;
                                                    back_col[l] = m.col;
                                                    set_pen_color(color::blue);

      fill_rectangle(a.col*square_width,a.row*square_width,square_width,square_w
idth);
                                                    {       if(maze[m.row][m.col - 1] != '#' && 
been_there[m.row][m.col - 1] != 1)
                                                    {       set_pen_color(color::white);
                                                            fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                            draw_grid(row_size, col_size, 
square_width);
                                                            back_row[l] = m.row;
                                                            back_col[l] = m.col;
                                                            m.col--;
                                                            set_pen_color(color::brown);
                                                            fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                            l++;
                                                            been_there[m.row][m.col] = 1; }
                                                    else if(maze[m.row][m.col + 1] != '#' 
&& been_there[m.row][m.col + 1] != 1)
                                                    {       set_pen_color(color::white);
                                                            fill_rectangle(m.col * 
square_width, m.row * square_width, square_width, square_width);
                                                            draw_grid(row_size, col_size, 
square_width);
                                                            back_row[l]=m.row;
                                                            back_col[l]=m.col;
                                                            m.col++;
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            l++;
                                                            been_there[m.row][m.col] = 1; }
                                                      else if(maze[m.row + 1][m.col] != '#'
&& been_there[m.row + 1][m.col] != 1)
                                                      {     set_pen_color(color::white);
                                                            fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                            draw_grid(row_size, col_size,
square_width);
                                                            back_row[l]=m.row;
                                                            back_col[l]=m.col;
                                                            m.row++;
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            l++;
                                                            been_there[m.row][m.col] = 1;}
                                                      else if(maze[m.row - 1][m.col] != '#'
&& been_there[m.row - 1][m.col] != 1)
                                                      {     set_pen_color(color::white);
                                                            fill_rectangle(m.col *
square_width, m.row * square_width, square_width, square_width);
                                                            draw_grid(row_size, col_size,
square_width);
                                                            back_row[l]=m.row;
                                                            back_col[l]=m.col;
                                                            m.row--;
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            l++;
                                                            been_there[m.row][m.col] = 1; }
                                                      else
                                                      {     set_pen_color(color::white);
                                                            fill_rectangle(m.col *
square_width +1, m.row * square_width+1, square_width-1, square_width-1);
                                                            l--;
                                                            m.row = back_row[l];
                                                            m.col = back_col[l];
                                                            set_pen_color(color::brown);

      fill_rectangle(m.col*square_width,m.row*square_width,square_width,square_w
idth);
                                                            been_there[m.row][m.col] = 1;}
}
                                                break; }
```

```
                                }
                        }
                }
        }
}


void draw_maze(string maze[], int const row_size, int const col_size, int const
square_width)
{       int r_width= row_size*square_width;
        int c_width= col_size*square_width;
        make_window(c_width,r_width);
        mazelab a, b, m;
        for(int i = 0; i < row_size; i++)
        {       for(int j = 0; j < col_size; j++)
                {       if(maze[i][j]=='#')
                        {       set_pen_color(color::grey);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width);}
                        else if(maze[i][j]=='~')
                        {       set_pen_color(color::white);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='+')
                        {       a.col = j;
                                a.row = i;
                                set_pen_color(color::blue);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='$')
                        {       b.col = j;
                                b.row = i;
                                set_pen_color(color::green);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                        else if(maze[i][j]=='E')
                        {       m.col = j;
                                m.row = i;
                                set_pen_color(color::brown);

        fill_rectangle(j*square_width,i*square_width,square_width,square_width); }
                }
        }
        draw_grid(row_size, col_size, square_width);
        move_robot(maze, a, b, m, square_width,row_size, col_size);
}

void read_maze()
{       int const row_size = 11;
        int const col_size = 31;
        int const square_width = 40;
        string maze[100];
        ifstream fin("C:/Users/sloanatkins/Desktop/maze.txt");
        if(fin.fail())
        {       cout << "Not available" << endl; }
        while(!fin.eof())
        {       for(int i = 0; i < 100; i++)
                {       fin >> maze[i]; } }
        fin.close();
        print_maze(maze, row_size, col_size);
        draw_maze(maze, row_size, col_size, square_width);
}
void main()
{       read_maze(); }
```



You have been Caught!!!

GAME OVER