Lab 12

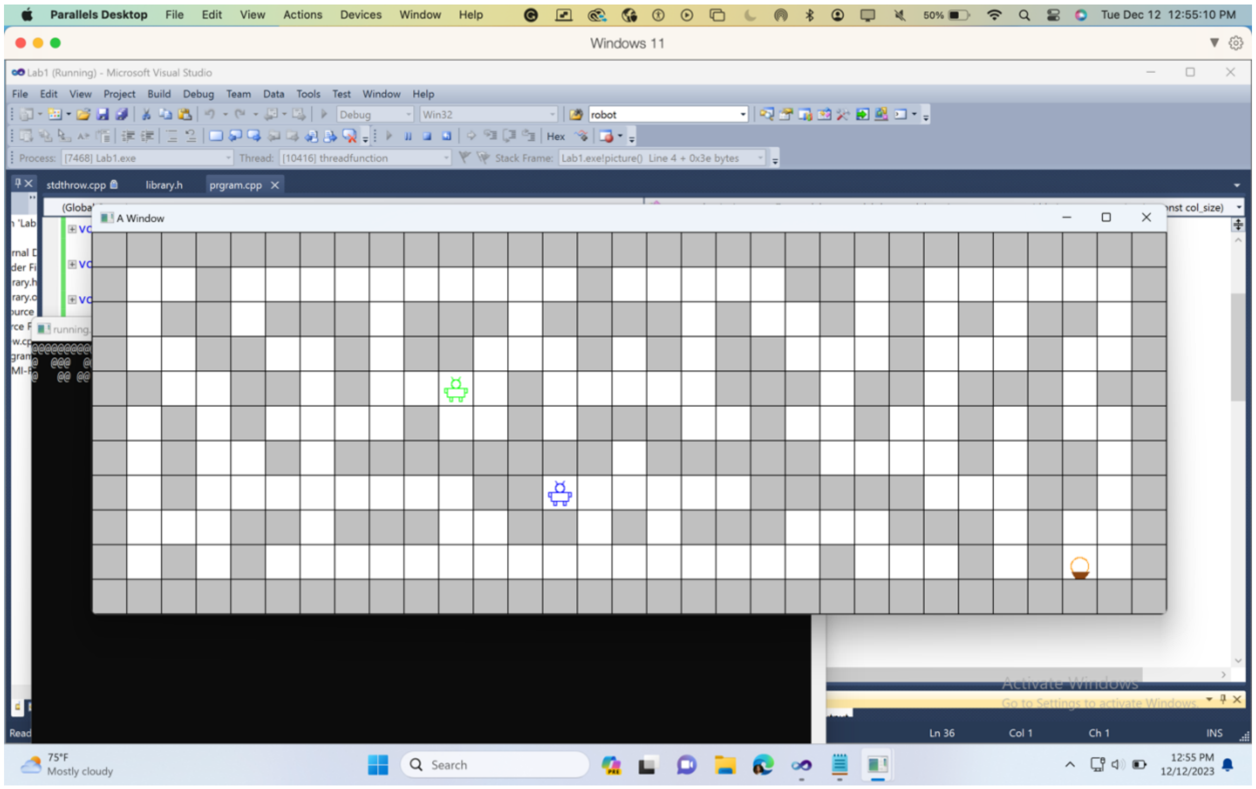
ECE 118 – Section R/RC
Lab on Wednesday at 5:05
Sloan Atkins

7. Better Graphics

```
#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 circle(const int N, int radius, int degrees)
{       if (N > 90 - degrees)
        {       const double pi = acos(-1.0);
    double x = (2*pi*radius)/360;
        set_pen_width(1);
        draw_distance(x);
        turn_right_by_degrees(1);
        circle(N-1, radius, degrees);} }

void circle2(const int N, int radius, int degrees)
{       if (N > 90 - degrees)
        {       set_pen_color(color::orange);
                const double pi = acos(-1.0);
                double x = (2*pi*radius)/360;
                set_pen_width(1);
                draw_distance(x);
                turn_right_by_degrees(1);
                circle(N-1, radius, degrees);}
}

void robot(int a, int b)
{
        set_heading_degrees(0);
        set_pen_color(color::blue);
        move_to(a*40+10,b*40+30);
        turn_right_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);
        turn_left_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);

        move_to(a*40+13,b*40+30);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+24,b*40+30);
        turn_left_by_degrees(180);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+10,b*40+20);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        move_to(a*40+30,b*40+20);
        draw_distance(3);
        turn_right_by_degrees(90);
        draw_distance(5);
        turn_right_by_degrees(90);
        draw_distance(3);

        move_to(a*40+20,b*40+20);
        circle(720, 5, 40);

        set_heading_degrees(180);

        move_to(a*40+15,b*40+7);
        turn_left_by_degrees(30);
        draw_distance(5);

        move_to(a*40+25,b*40+7);
        turn_right_by_degrees(60);
        draw_distance(5);

}

void gold(int a, int b)
{
        set_pen_color(color::yellow);
        set_heading_degrees(90);
        move_to(a*40+20,b*40+15);
        circle2(420,10,40);

        set_heading_degrees(0);
        set_pen_color(color::brown);
        move_to(a*40+15,b*40+40);
        note_position();
        turn_right_by_degrees(90);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(55);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(20);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(10);
        note_position();
        fill_shape();

}

void evil_robot(int a, int b)
{
        set_heading_degrees(0);
```

```
set_pen_color(color::green);
move_to(a*40+10,b*40+30);
turn_right_by_degrees(90);
draw_distance(20);
turn_left_by_degrees(90);
draw_distance(10);
turn_left_by_degrees(90);
draw_distance(20);
turn_left_by_degrees(90);
draw_distance(10);

move_to(a*40+13,b*40+30);
draw_distance(5);
turn_left_by_degrees(90);
draw_distance(3);
turn_left_by_degrees(90);
draw_distance(5);

move_to(a*40+24,b*40+30);
turn_left_by_degrees(180);
draw_distance(5);
turn_left_by_degrees(90);
draw_distance(3);
turn_left_by_degrees(90);
draw_distance(5);

move_to(a*40+10,b*40+20);
turn_left_by_degrees(90);
draw_distance(3);
turn_left_by_degrees(90);
draw_distance(5);
turn_left_by_degrees(90);
draw_distance(3);

move_to(a*40+30,b*40+20);
draw_distance(3);
turn_right_by_degrees(90);
draw_distance(5);
turn_right_by_degrees(90);
draw_distance(3);

move_to(a*40+20,b*40+20);
circle(720, 5, 40);

set_heading_degrees(180);

move_to(a*40+15,b*40+7);
turn_left_by_degrees(30);
draw_distance(5);

move_to(a*40+25,b*40+7);
turn_right_by_degrees(60);
```

```
        draw_distance(5);

}

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);
                                robot(j,i); }
                        else if(maze[i][j]=='$')
                        {       b.col = j;
                                b.row = i;
                                set_pen_color(color::green);
                                gold(j,i); }
                        else if(maze[i][j]=='E')
                        {       m.col = j;
                                m.row = i;
                                set_pen_color(color::brown);
                                evil_robot(j,i); }
                }
        }
        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(); }
```



8. Retracing Steps

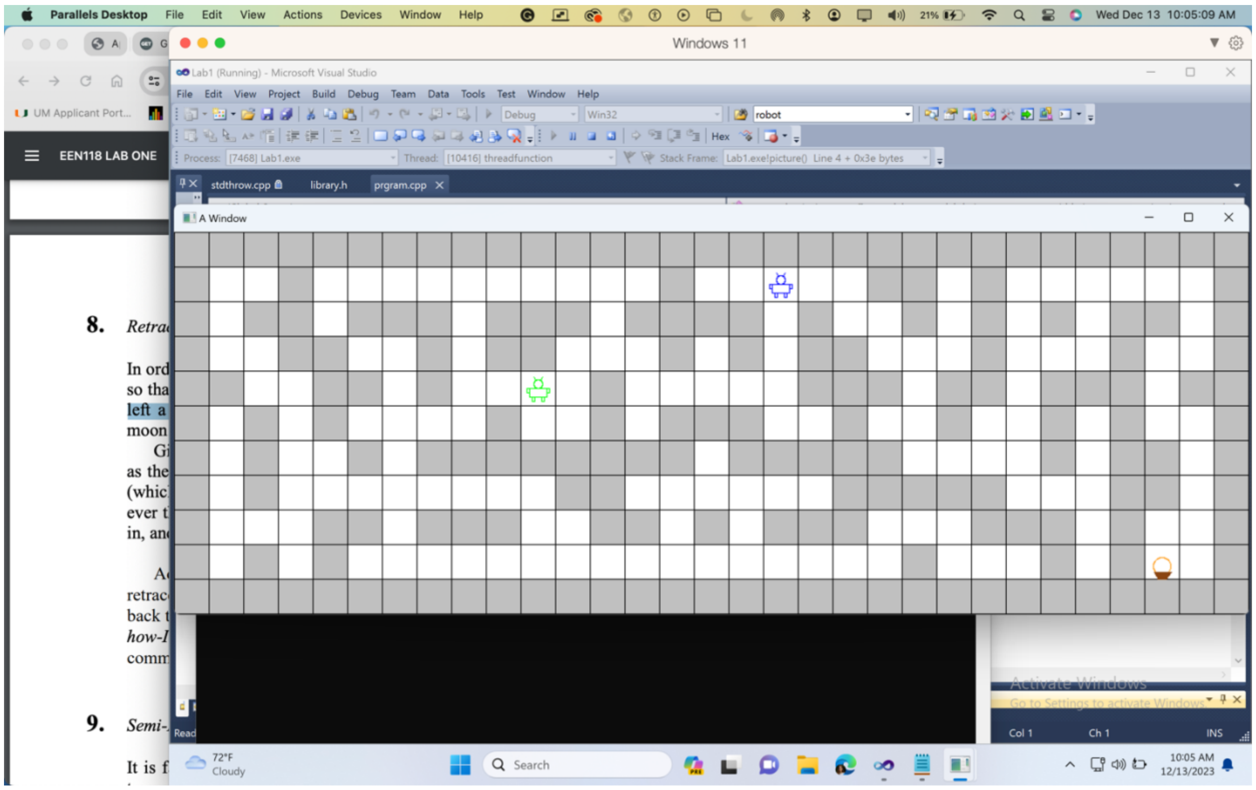
```
#include "library.h"

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 circle(const int N, int radius, int degrees)
{       if (N > 90 - degrees)
        {       const double pi = acos(-1.0);
    double x = (2*pi*radius)/360;
        set_pen_width(1);
        draw_distance(x);
        turn_right_by_degrees(1);
        circle(N-1, radius, degrees);} }

void circle2(const int N, int radius, int degrees)
{       if (N > 90 - degrees)
        {       set_pen_color(color::orange);
                const double pi = acos(-1.0);
                double x = (2*pi*radius)/360;
                set_pen_width(1);
                draw_distance(x);
                turn_right_by_degrees(1);
                circle(N-1, radius, degrees);}
}

void robot(int a, int b)
{
        set_heading_degrees(0);
        set_pen_color(color::blue);
        move_to(a*40+10,b*40+30);
        turn_right_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);
        turn_left_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);

        move_to(a*40+13,b*40+30);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+24,b*40+30);
        turn_left_by_degrees(180);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+10,b*40+20);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);

        move_to(a*40+30,b*40+20);
        draw_distance(3);
        turn_right_by_degrees(90);
        draw_distance(5);
        turn_right_by_degrees(90);
        draw_distance(3);

        move_to(a*40+20,b*40+20);
        circle(720, 5, 40);

        set_heading_degrees(180);

        move_to(a*40+15,b*40+7);
        turn_left_by_degrees(30);
        draw_distance(5);

        move_to(a*40+25,b*40+7);
        turn_right_by_degrees(60);
        draw_distance(5);

}

void gold(int a, int b)
```

```
{
        set_pen_color(color::yellow);
        set_heading_degrees(90);
        move_to(a*40+20,b*40+15);
        circle2(420,10,40);

        set_heading_degrees(0);
        set_pen_color(color::brown);
        move_to(a*40+15,b*40+40);
        note_position();
        turn_right_by_degrees(90);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(55);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(20);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(10);
        note_position();
        fill_shape();

}

void move_robot(string maze[], mazelab a, mazelab b, 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);
                gold(b.col,b.row);
                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);
                            robot(a.col,a.row);
                            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++;
                            robot(a.col,a.row);
                            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--;
                            robot(a.col,a.row);
                            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++;
                            robot(a.col,a.row);
                            i++; }}

            if(c == 'b')
            {       if(maze[a.row+1][a.col] != '#')
```

```
                    {     set_pen_color(color::white);
                          fill_rectangle(a.col * square_width +1, a.row *
square_width+1, square_width-1, square_width-1);
                          l--;
                          a.row = back_row[l];
                          a.col = back_col[l];
                          robot(a.col,a.row);
                          been_there[a.row][a.col] = 1;}}
      }
}

void evil_robot(int a, int b)
{
      set_heading_degrees(0);
      set_pen_color(color::green);
      move_to(a*40+10,b*40+30);
      turn_right_by_degrees(90);
      draw_distance(20);
      turn_left_by_degrees(90);
      draw_distance(10);
      turn_left_by_degrees(90);
      draw_distance(20);
      turn_left_by_degrees(90);
      draw_distance(10);

      move_to(a*40+13,b*40+30);
      draw_distance(5);
      turn_left_by_degrees(90);
      draw_distance(3);
      turn_left_by_degrees(90);
      draw_distance(5);

      move_to(a*40+24,b*40+30);
      turn_left_by_degrees(180);
      draw_distance(5);
      turn_left_by_degrees(90);
      draw_distance(3);
      turn_left_by_degrees(90);
      draw_distance(5);

      move_to(a*40+10,b*40+20);
      turn_left_by_degrees(90);
      draw_distance(3);
      turn_left_by_degrees(90);
      draw_distance(5);
      turn_left_by_degrees(90);
      draw_distance(3);

      move_to(a*40+30,b*40+20);
      draw_distance(3);
      turn_right_by_degrees(90);
      draw_distance(5);
```

```
        turn_right_by_degrees(90);
        draw_distance(3);

        move_to(a*40+20,b*40+20);
        circle(720, 5, 40);

        set_heading_degrees(180);

        move_to(a*40+15,b*40+7);
        turn_left_by_degrees(30);
        draw_distance(5);

        move_to(a*40+25,b*40+7);
        turn_right_by_degrees(60);
        draw_distance(5);

}

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);
                                robot(j,i); }
                        else if(maze[i][j]=='$')
                        {       b.col = j;
                                b.row = i;
                                set_pen_color(color::green);
                                gold(j,i); }
                        else if(maze[i][j]=='E')
                        {       m.col = j;
                                m.row = i;
                                set_pen_color(color::brown);
                                evil_robot(j,i); }
                }
        }
        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(); }
```



Parallels Desktop File Edit View Actions Devices Window Help
Wed Dec 13 10:05:09 AM
Windows 11
Lab1 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test Window Help
Debug
Win32
robot
Process: [7468] Lab1.exe
Thread: [10416] threadfunction
Stack Frame: Lab1.exe!picture() Line 4 + 0x3e bytes
stdthrow.cpp
library.h
prgram.cpp
A Window
EEN118 LAB ONE
UM Applicant Port...
Activate Windows
Go to Settings to activate Windows.
Col 1
Ch 1
INS
72°F Cloudy
Search
10:05 AM 12/13/2023

## 9. Semi-Intelligent Behaviour

```
#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 circle(const int N, int radius, int degrees)
{       if (N > 90 - degrees)
        {       const double pi = acos(-1.0);
      double x = (2*pi*radius)/360;
        set_pen_width(1);
        draw_distance(x);
        turn_right_by_degrees(1);
        circle(N-1, radius, degrees);} }

void circle2(const int N, int radius, int degrees)
```

```
{       if (N > 90 - degrees)
        {       set_pen_color(color::orange);
                const double pi = acos(-1.0);
                double x = (2*pi*radius)/360;
                set_pen_width(1);
                draw_distance(x);
                turn_right_by_degrees(1);
                circle(N-1, radius, degrees);}
}

void robot(int a, int b)
{
        set_heading_degrees(0);
        set_pen_color(color::blue);
        move_to(a*40+10,b*40+30);
        turn_right_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);
        turn_left_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);

        move_to(a*40+13,b*40+30);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+24,b*40+30);
        turn_left_by_degrees(180);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+10,b*40+20);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);

        move_to(a*40+30,b*40+20);
        draw_distance(3);
        turn_right_by_degrees(90);
        draw_distance(5);
        turn_right_by_degrees(90);
        draw_distance(3);

        move_to(a*40+20,b*40+20);
        circle(720, 5, 40);

        set_heading_degrees(180);

        move_to(a*40+15,b*40+7);
        turn_left_by_degrees(30);
        draw_distance(5);

        move_to(a*40+25,b*40+7);
        turn_right_by_degrees(60);
        draw_distance(5);

}

void gold(int a, int b)
{
        set_pen_color(color::yellow);
        set_heading_degrees(90);
        move_to(a*40+20,b*40+15);
        circle2(420,10,40);

        set_heading_degrees(0);
        set_pen_color(color::brown);
        move_to(a*40+15,b*40+40);
        note_position();
        turn_right_by_degrees(90);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(55);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(20);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(10);
        note_position();
        fill_shape();

}

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

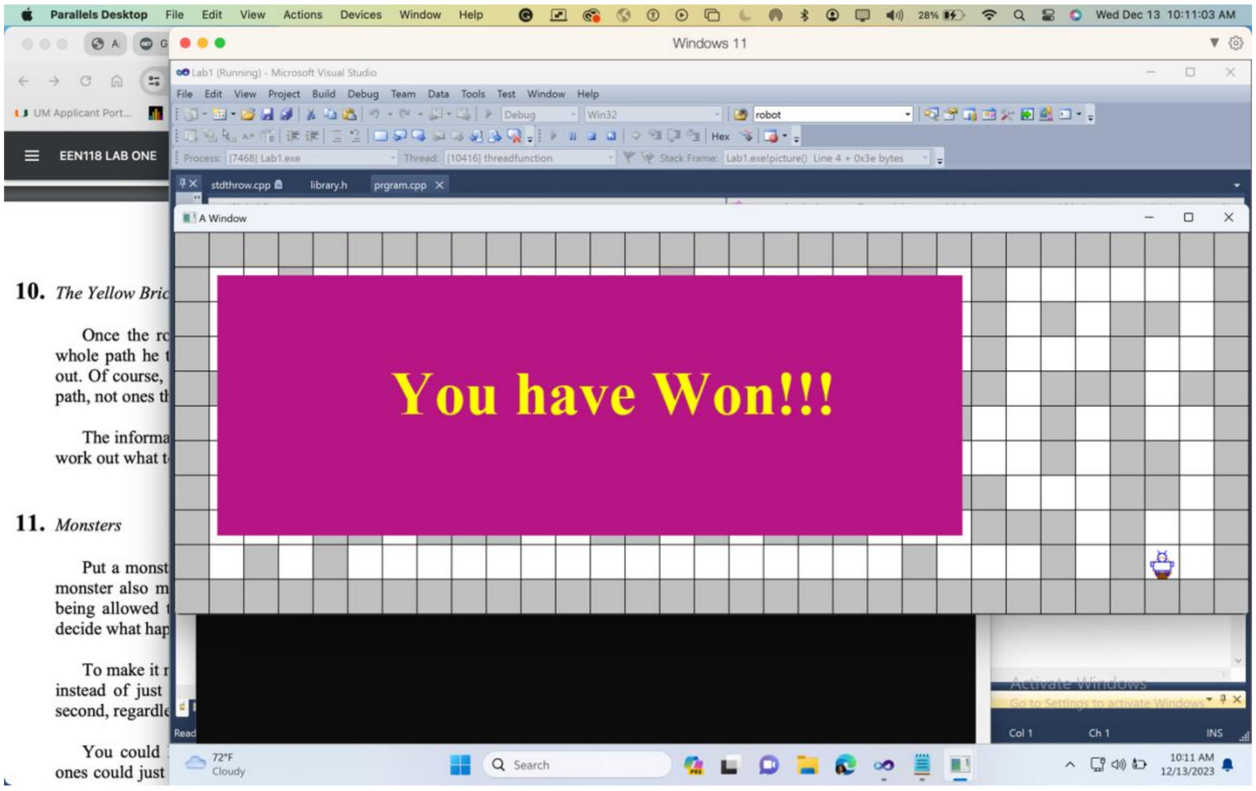
```
                        fill_rectangle(a.col * square_width, a.row *
square_width, square_width, square_width);
                        draw_grid(row_size, col_size, square_width);
                        back_row[i]=a.row;
                        back_col[i]=a.col;
                        a.row++;
                        robot(a.col,a.row);
                        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--;
                                    robot(a.col,a.row);
                                    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++;
                                                robot(a.col,a.row);
                                                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++;
                                                robot(a.col,a.row);
                                                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--;
                                                robot(a.col,a.row);
                                                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];
                                                robot(a.col,a.row);
                                                break;}
                                }
                        }
                }
        }
}
```

```
void evil_robot(int a, int b)
{
	set_heading_degrees(0);
	set_pen_color(color::green);
	move_to(a*40+10,b*40+30);
	turn_right_by_degrees(90);
	draw_distance(20);
	turn_left_by_degrees(90);
	draw_distance(10);
	turn_left_by_degrees(90);
	draw_distance(20);
	turn_left_by_degrees(90);
	draw_distance(10);

	move_to(a*40+13,b*40+30);
	draw_distance(5);
	turn_left_by_degrees(90);
	draw_distance(3);
	turn_left_by_degrees(90);
	draw_distance(5);

	move_to(a*40+24,b*40+30);
	turn_left_by_degrees(180);
	draw_distance(5);
	turn_left_by_degrees(90);
	draw_distance(3);
	turn_left_by_degrees(90);
	draw_distance(5);

	move_to(a*40+10,b*40+20);
	turn_left_by_degrees(90);
	draw_distance(3);
	turn_left_by_degrees(90);
	draw_distance(5);
	turn_left_by_degrees(90);
	draw_distance(3);

	move_to(a*40+30,b*40+20);
	draw_distance(3);
	turn_right_by_degrees(90);
	draw_distance(5);
	turn_right_by_degrees(90);
	draw_distance(3);

	move_to(a*40+20,b*40+20);
	circle(720, 5, 40);

	set_heading_degrees(180);

	move_to(a*40+15,b*40+7);
	turn_left_by_degrees(30);
```

```
        draw_distance(5);

        move_to(a*40+25,b*40+7);
        turn_right_by_degrees(60);
        draw_distance(5);

}

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);
                                robot(j,i); }
                        else if(maze[i][j]=='$')
                        {       b.col = j;
                                b.row = i;
                                set_pen_color(color::green);
                                gold(j,i); }
                        else if(maze[i][j]=='E')
                        {       m.col = j;
                                m.row = i;
                                set_pen_color(color::brown);
                                evil_robot(j,i); }
                }
        }
        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(); }
```


Parallels Desktop File Edit View Actions Devices Window Help
Wed Dec 13 10:11:03 AM
Windows 11
Lab1 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test Window Help
Debug
Win32
robot
Process: [7468] Lab1.exe
Thread: [10416] threadfunction
Stack Frame: Lab1.exe!picture() Line 4 + 0x3e bytes
stdthrow.cpp
library.h
prgram.cpp
A Window
You have Won!!!
UM Applicant Port...
EEN118 LAB ONE
Activate Windows
Go to Settings to activate Windows.
Col 1
Ch 1
INS
72°F
Cloudy
Search
10:11 AM
12/13/2023

10. *The Yellow Bric*

Once the r
whole path he t
out. Of course,
path, not ones t

The informa
work out what t

11. *Monsters*

Put a monst
monster also m
being allowed t
decide what hap

To make it r
instead of just
second, regardle

You could
ones could just

10. The Yellow Brick Road

```
#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 circle(const int N, int radius, int degrees)
{       if (N > 90 - degrees)
        {       const double pi = acos(-1.0);
     double x = (2*pi*radius)/360;
        set_pen_width(1);
        draw_distance(x);
        turn_right_by_degrees(1);
        circle(N-1, radius, degrees);} }
```

```
void circle2(const int N, int radius, int degrees)
{      if (N > 90 - degrees)
       {      set_pen_color(color::orange);
              const double pi = acos(-1.0);
              double x = (2*pi*radius)/360;
              set_pen_width(1);
              draw_distance(x);
              turn_right_by_degrees(1);
              circle(N-1, radius, degrees);}
}

void robot(int a, int b)
{
       set_heading_degrees(0);
       set_pen_color(color::blue);
       move_to(a*40+10,b*40+30);
       turn_right_by_degrees(90);
       draw_distance(20);
       turn_left_by_degrees(90);
       draw_distance(10);
       turn_left_by_degrees(90);
       draw_distance(20);
       turn_left_by_degrees(90);
       draw_distance(10);

       move_to(a*40+13,b*40+30);
       draw_distance(5);
       turn_left_by_degrees(90);
       draw_distance(3);
       turn_left_by_degrees(90);
       draw_distance(5);

       move_to(a*40+24,b*40+30);
       turn_left_by_degrees(180);
       draw_distance(5);
       turn_left_by_degrees(90);
       draw_distance(3);
       turn_left_by_degrees(90);
       draw_distance(5);

       move_to(a*40+10,b*40+20);
       turn_left_by_degrees(90);
       draw_distance(3);
       turn_left_by_degrees(90);
       draw_distance(5);
       turn_left_by_degrees(90);
       draw_distance(3);

       move_to(a*40+30,b*40+20);
       draw_distance(3);
       turn_right_by_degrees(90);
       draw_distance(5);
```

```
        turn_right_by_degrees(90);
        draw_distance(3);

        move_to(a*40+20,b*40+20);
        circle(720, 5, 40);

        set_heading_degrees(180);

        move_to(a*40+15,b*40+7);
        turn_left_by_degrees(30);
        draw_distance(5);

        move_to(a*40+25,b*40+7);
        turn_right_by_degrees(60);
        draw_distance(5);

}

void gold(int a, int b)
{
        set_pen_color(color::yellow);
        set_heading_degrees(90);
        move_to(a*40+20,b*40+15);
        circle2(420,10,40);

        set_heading_degrees(0);
        set_pen_color(color::brown);
        move_to(a*40+15,b*40+40);
        note_position();
        turn_right_by_degrees(90);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(55);
        draw_distance(10);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(20);
        note_position();
        turn_left_by_degrees(125);
        draw_distance(10);
        note_position();
        fill_shape();

}

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();
```

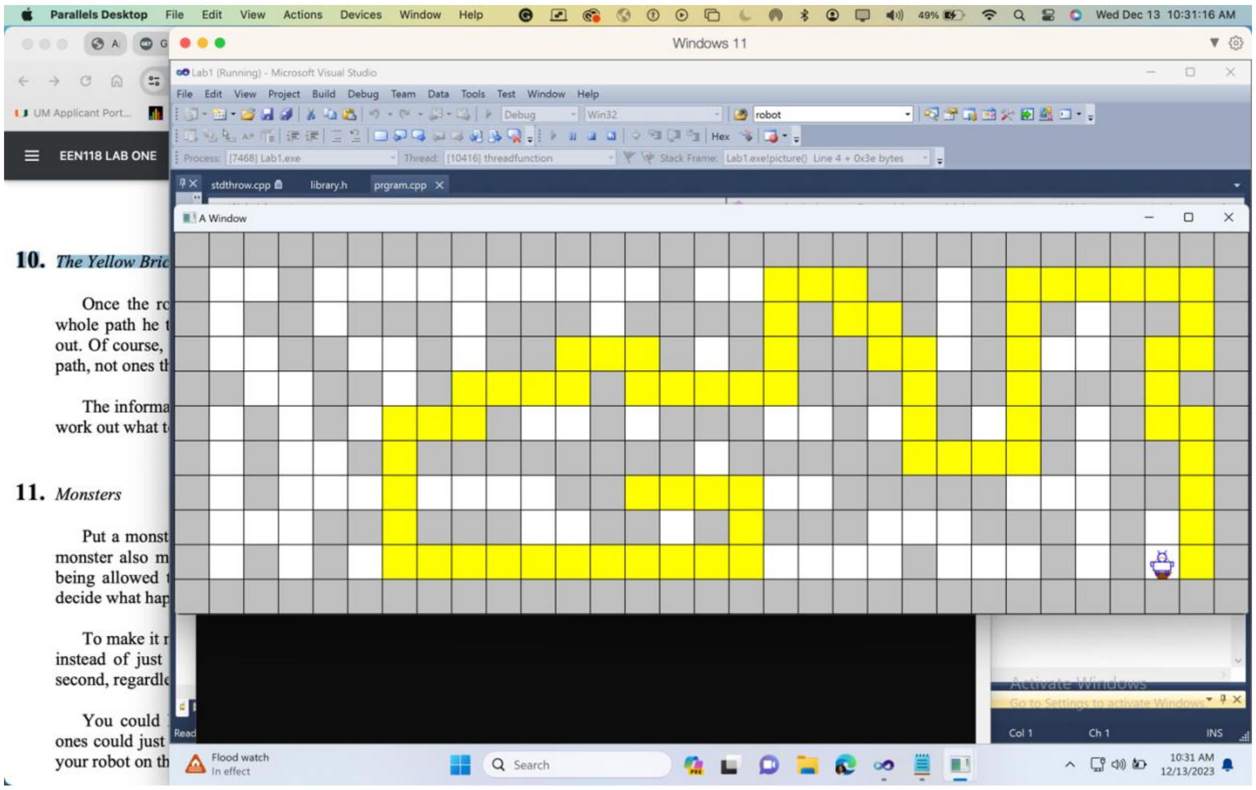
```
            if(c == 'x')
                  exit(1);
            if(a.row == b.row && a.col == b.col)
            {     draw_grid(row_size, col_size, square_width);
                  for(int j = 0; back_row[j] >= 1 && back_row[j] <= 9; j++)
                  {     int row = back_row[j];
                        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::yellow);
                        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--;
                        robot(a.col,a.row);
                        i++; }
            }
            if(c == -89)
            {     if(maze[a.row][a.col+1] != '#')
                  {     set_pen_color(color::yellow);
                        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++;
                        robot(a.col,a.row);
                        i++; }
            }
            if(c == -90)
            {     if(maze[a.row-1][a.col] != '#')
                  {     set_pen_color(color::yellow);
                        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--;
                        robot(a.col,a.row);
                        i++; }
            }
            if(c == -88)
            {     if(maze[a.row+1][a.col] != '#')
```

```
                    {       set_pen_color(color::yellow);
                            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++;
                            robot(a.col,a.row);
                            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::yellow);
                                            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--;
                                            robot(a.col,a.row);
                                            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::yellow);
                                            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++;
                                                robot(a.col,a.row);
                                                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::yellow);
                                                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++;
                                                robot(a.col,a.row);
                                                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::yellow);
                                                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--;
                                                robot(a.col,a.row);
                                                i++;
                                                been_there[a.row][a.col] = 1;
                                                break; }
                                        else if(i > 0)
                                        {       set_pen_color(color::yellow);
                                                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];
                                                robot(a.col,a.row);
                                                break;}
                                }
                        }
                }
        }
}
```

```
void evil_robot(int a, int b)
{
	set_heading_degrees(0);
	set_pen_color(color::green);
	move_to(a*40+10,b*40+30);
	turn_right_by_degrees(90);
	draw_distance(20);
	turn_left_by_degrees(90);
	draw_distance(10);
	turn_left_by_degrees(90);
	draw_distance(20);
	turn_left_by_degrees(90);
	draw_distance(10);

	move_to(a*40+13,b*40+30);
	draw_distance(5);
	turn_left_by_degrees(90);
	draw_distance(3);
	turn_left_by_degrees(90);
	draw_distance(5);

	move_to(a*40+24,b*40+30);
	turn_left_by_degrees(180);
	draw_distance(5);
	turn_left_by_degrees(90);
	draw_distance(3);
	turn_left_by_degrees(90);
	draw_distance(5);

	move_to(a*40+10,b*40+20);
	turn_left_by_degrees(90);
	draw_distance(3);
	turn_left_by_degrees(90);
	draw_distance(5);
	turn_left_by_degrees(90);
	draw_distance(3);

	move_to(a*40+30,b*40+20);
	draw_distance(3);
	turn_right_by_degrees(90);
	draw_distance(5);
	turn_right_by_degrees(90);
	draw_distance(3);

	move_to(a*40+20,b*40+20);
	circle(720, 5, 40);

	set_heading_degrees(180);

	move_to(a*40+15,b*40+7);
	turn_left_by_degrees(30);
	draw_distance(5);

	move_to(a*40+25,b*40+7);
	turn_right_by_degrees(60);
	draw_distance(5);

}

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);
				robot(j,i); }
			else if(maze[i][j]=='$')
			{	b.col = j;
				b.row = i;
				set_pen_color(color::green);
				gold(j,i); }
			else if(maze[i][j]=='E')
			{	m.col = j;
				m.row = i;
				set_pen_color(color::brown);
				evil_robot(j,i); }
		}
	}
	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(); }
```



11. Monsters

```
#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 circle(const int N, int radius, int degrees)
{     if (N > 90 - degrees)
      {     const double pi = acos(-1.0);
    double x = (2*pi*radius)/360;
      set_pen_width(1);
      draw_distance(x);
      turn_right_by_degrees(1);
      circle(N-1, radius, degrees);} }

void circle2(const int N, int radius, int degrees)
{     if (N > 90 - degrees)
      {     set_pen_color(color::orange);
            const double pi = acos(-1.0);
            double x = (2*pi*radius)/360;
            set_pen_width(1);
            draw_distance(x);
            turn_right_by_degrees(1);
            circle(N-1, radius, degrees);}
}

void robot(int a, int b)
```

```
{
        set_heading_degrees(0);
        set_pen_color(color::blue);
        move_to(a*40+10,b*40+30);
        turn_right_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);
        turn_left_by_degrees(90);
        draw_distance(20);
        turn_left_by_degrees(90);
        draw_distance(10);

        move_to(a*40+13,b*40+30);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+24,b*40+30);
        turn_left_by_degrees(180);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+10,b*40+20);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);

        move_to(a*40+30,b*40+20);
        draw_distance(3);
        turn_right_by_degrees(90);
        draw_distance(5);
        turn_right_by_degrees(90);
        draw_distance(3);

        move_to(a*40+20,b*40+20);
        circle(720, 5, 40);

        set_heading_degrees(180);

        move_to(a*40+15,b*40+7);
        turn_left_by_degrees(30);
        draw_distance(5);
```

```
      move_to(a*40+25,b*40+7);
      turn_right_by_degrees(60);
      draw_distance(5);

}

void gold(int a, int b)
{
      set_pen_color(color::yellow);
      set_heading_degrees(90);
      move_to(a*40+20,b*40+15);
      circle2(420,10,40);

      set_heading_degrees(0);
      set_pen_color(color::brown);
      move_to(a*40+15,b*40+40);
      note_position();
      turn_right_by_degrees(90);
      draw_distance(10);
      note_position();
      turn_left_by_degrees(55);
      draw_distance(10);
      note_position();
      turn_left_by_degrees(125);
      draw_distance(20);
      note_position();
      turn_left_by_degrees(125);
      draw_distance(10);
      note_position();
      fill_shape();

}

void evil_robot(int a, int b)
{
      set_heading_degrees(0);
      set_pen_color(color::green);
      move_to(a*40+10,b*40+30);
      turn_right_by_degrees(90);
      draw_distance(20);
      turn_left_by_degrees(90);
      draw_distance(10);
      turn_left_by_degrees(90);
      draw_distance(20);
      turn_left_by_degrees(90);
      draw_distance(10);

      move_to(a*40+13,b*40+30);
      draw_distance(5);
      turn_left_by_degrees(90);
      draw_distance(3);
      turn_left_by_degrees(90);
```

```
        draw_distance(5);

        move_to(a*40+24,b*40+30);
        turn_left_by_degrees(180);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);

        move_to(a*40+10,b*40+20);
        turn_left_by_degrees(90);
        draw_distance(3);
        turn_left_by_degrees(90);
        draw_distance(5);
        turn_left_by_degrees(90);
        draw_distance(3);

        move_to(a*40+30,b*40+20);
        draw_distance(3);
        turn_right_by_degrees(90);
        draw_distance(5);
        turn_right_by_degrees(90);
        draw_distance(3);

        move_to(a*40+20,b*40+20);
        circle(720, 5, 40);

        set_heading_degrees(180);

        move_to(a*40+15,b*40+7);
        turn_left_by_degrees(30);
        draw_distance(5);

        move_to(a*40+25,b*40+7);
        turn_right_by_degrees(60);
        draw_distance(5);

}

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();
                gold(b.col,b.row);
                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--;
                            robot(a.col,a.row);
                            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--;
                                            evil_robot(m.col,m.row);
                                            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++;
                                        evil_robot(m.col,m.row);
                                        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++;
                                        evil_robot(m.col,m.row);
                                        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--;
                                        evil_robot(m.col,m.row);
                                        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];
                                        evil_robot(m.col,m.row); } }

                        }
                }
                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++;
                            robot(a.col,a.row);
                            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--;
                                            evil_robot(m.col,m.row);
                                            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++;
                                            evil_robot(m.col,m.row);
                                            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++;
                                            evil_robot(m.col,m.row);
                                            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--;
                                    evil_robot(m.col,m.row);
                                    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];
                                    evil_robot(m.col,m.row); } }
                  }
            }
            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--;
                        robot(a.col,a.row);
                        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--;
                                    evil_robot(m.col,m.row);
                                    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++;
                                      evil_robot(m.col,m.row);
                                      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++;
                                      evil_robot(m.col,m.row);
                                      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--;
                                      evil_robot(m.col,m.row);
                                      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];
                                      evil_robot(m.col,m.row);} }}
            }
            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++;
                         robot(a.col,a.row);
                         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--;
                                       evil_robot(m.col,m.row);
                                       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++;
                                       evil_robot(m.col,m.row);
                                       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++;
                                       evil_robot(m.col,m.row);
                                       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--;
                                        evil_robot(m.col,m.row);
                                        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];
                                        evil_robot(m.col,m.row); } }
                                }
                        }
                if(c == 'a')
                {       while(true)
                        {       gold(b.col,b.row);
                                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--;
                                        robot(a.col,a.row);
                                        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--;
                                                        evil_robot(m.col,m.row);
                                                        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++;
                                                        evil_robot(m.col,m.row);
                                                        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++;
                                                        evil_robot(m.col,m.row);
                                                        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--;
                                                    evil_robot(m.col,m.row);
                                                    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];
                                                    evil_robot(m.col,m.row);
                                                    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++;
                                        robot(a.col,a.row);
                                        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--;
                                                    evil_robot(m.col,m.row);
                                                    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++;
                                                evil_robot(m.col,m.row);
                                                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++;
                                                evil_robot(m.col,m.row);
                                                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--;
                                                evil_robot(m.col,m.row);
                                                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];
                                                evil_robot(m.col,m.row);
                                                been_there[m.row][m.col] = 1;}
}
                                break; }
```

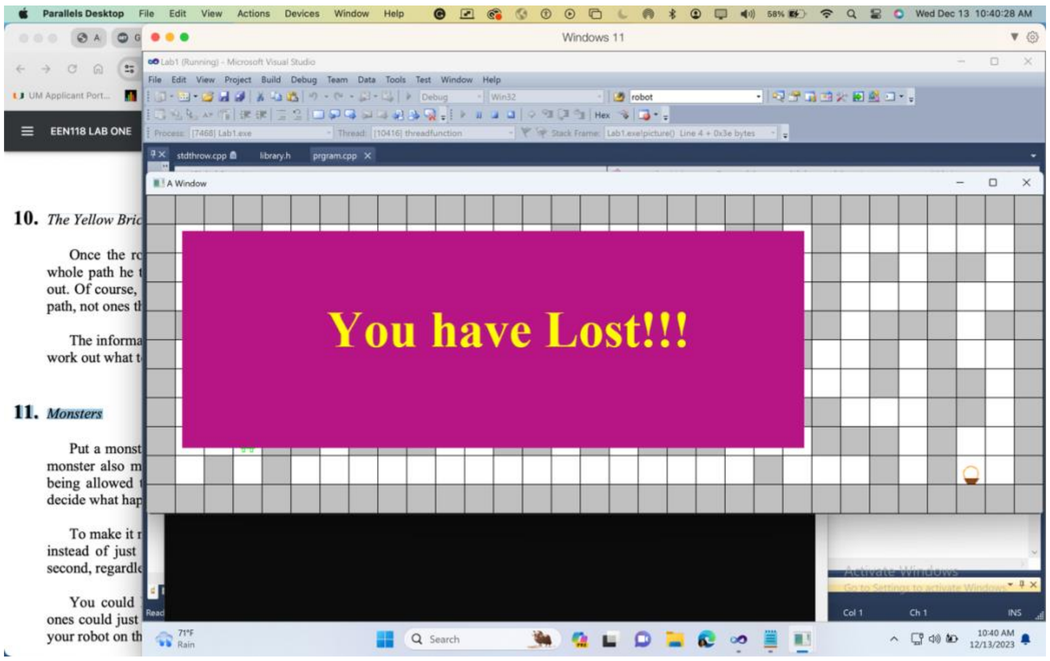
```
                                        else if(maze[a.row + 1][a.col] != '#' &&
been_there[a.row + 1][a.col] != 1)
                                   {     set_pen_color(color::white);
                                        fill_rectangle(a.col * square_width, a.row
* square_width, square_width,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

```
                                                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--;
                                                evil_robot(m.col,m.row);
                                                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];
                                                evil_robot(m.col,m.row);
                                                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--;
                                      robot(a.col,a.row);
                                      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--;
                                                evil_robot(m.col,m.row);
```

```
                                                                    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++;
                                                                    evil_robot(m.col,m.row);
                                                                    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++;
                                                                    evil_robot(m.col,m.row);
                                                                    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--;
                                                                    evil_robot(m.col,m.row);
                                                                    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];
                                                                    evil_robot(m.col,m.row);
                                                                    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;
                                        robot(a.col,a.row);
                                        {     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--;
                                                    evil_robot(m.col,m.row);
                                                    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++;
                                                    evil_robot(m.col,m.row);
                                                    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++;
                                                    evil_robot(m.col,m.row);
                                                    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--;
                                                                evil_robot(m.col,m.row);
                                                                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];
                                                                evil_robot(m.col,m.row);
                                                                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);
                                robot(j,i); }
                        else if(maze[i][j]=='$')
                        {       b.col = j;
                                b.row = i;
                                set_pen_color(color::green);
                gold(j,i); }
            else if(maze[i][j]=='E')
            {   m.col = j;
                m.row = i;
                set_pen_color(color::brown);
                evil_robot(j,i); }
        }
    }
    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 Lost!!!