```cpp
 1: #include <BearLibTerminal.h>
 2: #include <cmath>
 3: #include <iostream>
 4: #include <cstdlib>
 5: #include <ctime>
 6: #include <fstream>
 7: using namespace std;
 8: #include "gooseEscapeUtil.hpp"
 9: #include "gooseEscapeActors.hpp"
10: #include "gooseEscapeConsole.hpp"
11: #include "gooseEscapeGamePlay.hpp"
12:
13: //set up the console.   Don't modify this line!
14: Console out;
15:
16:
17: const int RANDOM=21;
18: const int MONSTER_X=70;
19: const int MONSTER_Y=20;
20:
21: int random_position_generator(int rand_num)
22: {
23:     int r_num=rand_num;
24:     srand(time(0));
25:     int num=(rand () % r_num)+1;
26:     return num;
27: }
28:
29: int main()
30: {
31:     int x=0,y=0;
32:     //Set up the window.  Don't edit these two lines
33:     terminal_open();
34:     terminal_set(SETUP_MESSAGE);
35:
36:
37: /*
38: The code below provides a skeleton of the game play.  You will need to
39: write code for setting up the game board, and playing the game itself.
40: You can modify the code given as needed.
41:
42: Call the functions that you have written in the game play file, and that
43: you have added to the Actor class.
44: */
45:
46:     //make the player
47:
48:     x=random_position_generator(80);
49:     y=random_position_generator(21);
50:     Actor player(PLAYER_CHAR, x,y);
51: /* random function called to assign different starting
52: positions each time the game is played.*/
53:
54:     //make the monster
55:     Actor monster(MONSTER_CHAR, MONSTER_X,MONSTER_Y);
```

```cpp
56:
57:        // Declare the array that will hold the game board "map"
58:        int game_world[NUM_BOARD_X][NUM_BOARD_Y]={0};
59: /*
60: Initiallize locations in the game board to have game features.  What if you
61: have man things to add to the game board?  Should you use a loop?  Does it
62: make sense to store this information in a file?  Should this code be a
63: function as well?
64: */
65: //we are using files to input the walls and the power up points into the game
66:        ifstream fin_wall("wall.txt");
67:        ifstream fin_power("power.txt");
68:        if(!fin_wall||!fin_power)
69:        {
70:            cout<<"ERROR IN FILE !!!";
71:            return 1;
72:        }
73:
74:        /* game map location */
75:        /* game map location */
76:        game_world[WIN][WIN] = WINNER;
77:
78:        // Call the function to print the game board
79:        setup(fin_wall,fin_power,game_world);
80:        // Printing the instructions
81:        out.writeLine("Escape the Goose! " + monster.get_location_string());
82:        out.writeLine("Use the arrow keys to move");
83:        out.writeLine("If the goose catches you, you lose!");
84:        out.writeLine("Be careful! Sometimes the goose can jump walls!");
85:
86: /*
87: This is the main game loop.  It continues to let the player give input
88: as long as they do not press escape or close, they are not captured by
89: the goose, and they didn't reach the win tile
90: */
91: /*
92: All key presses start with "TK_" then the character.  So "TK_A" is the "a"
93: key being pressed.
94: */
95:        int keyEntered = TK_A;
96:        int previous_step = TK_A;
97:        // can be any valid value that is not ESCAPE or CLOSE
98:
99:        while(keyEntered != TK_ESCAPE && keyEntered != TK_CLOSE
100:        && !captured(player,monster) &&
101:        game_world[player.get_x()][player.get_y()]!=WINNER)
102:        {
103:            // get player key press
104:            keyEntered = terminal_read();
105:
106:            if (keyEntered != TK_ESCAPE && keyEntered != TK_CLOSE)
107:            {
108:                if (keyEntered==TK_P && player.power_up())
109:                {
110:                    s_power(previous_step,player,game_world);
```

```
111:                }
112:                else
113:                {
114:                    // move the player, you can modify this function
115:                        movePlayer(keyEntered,player,game_world);
116:                }
117:                // call the goose's chase function
118:                move_the_Goose(player,monster,game_world);
119:                // call other functions to do stuff?
120:                if (!player.power_up())
121:                    power_point_checker(player,monster,game_world);
122:            }
123:
124:        if (keyEntered == TK_UP || keyEntered == TK_DOWN||
125:            keyEntered == TK_LEFT||keyEntered == TK_RIGHT)
126:        previous_step=keyEntered;
127:    }
128:
129:    if (keyEntered != TK_ESCAPE)
130:    {
131:        //once we're out of the loop, the game is over
132:        out.writeLine("Game Ended");
133:
134:        if(captured(player,monster))
135:            out.writeLine("The Goose have caught you!...Attempt Failed !!!");
136:        else if (game_world[player.get_x()][player.get_y()]==WINNER)
137:            out.writeLine("You have Out smarted the goose and reached a"
138:                            " safe spot!!! Congratulations!!!");
139:        //out.writeLine("Game has ended");
140:         while (terminal_read() != TK_CLOSE);
141:
142: // output why:  did the goose get us, or did we get to the win location?
143:    }    // Wait until user closes the window
144:
145:
146:
147:    //game is done, close it
148:    terminal_close();
149: }
```