

```

1: #include <iostream>
2: #include <cmath>
3: using namespace std;
4: #include <BearLibTerminal.h>
5: #include "gooseEscapeUtil.hpp"
6: #include "gooseEscapeActors.hpp"
7: #include "gooseEscapeConsole.hpp"
8: #include "gooseEscapeGamePlay.hpp"
9:
10: extern Console out;
11: /*
12:  With graphics, screens are given an x,y coordinate system with the o
13:  in the upper left corner. So it means the coordinate axes are:
14:  ----->    x direction
15:  /
16:  /
17:  /
18:  /
19:  /
20: V
21:
22: y direction
23: */
24: const int COL=12;
25: const int COL2=24;
26: const int MAN_STEP=1;
27: const int GOOSE_STEP=3;
28: /*
29:     Print the game world
30:
31:     The functions should draw characters to present features of the
32:     board, e.g. win location, obstacles, power ups
33: */
34:
35: // print the game board function
36: void print_board()
37: {
38:     for(int wall=COL;wall<=COL2;wall++)
39:     {
40:         terminal_put(wall,COL,WALL_CHAR);
41:     }
42:     terminal_put(WIN,WIN,WIN_CHAR);
43: }
44:
45: /*
46:     Do something when the goose captures the player
47:
48:     If you want to attack or something else, this is the function yo
49:     need to change. For example, maybe the two touch each other and

```

```

50:      then fight. You could add a health to the Actor class that is
51:      updated. Run, use weapons, it's up to you!
52:  */
53:
54:  bool captured(Actor const & player, Actor const & monster)
55:  {
56:      return (player.get_x() == monster.get_x()
57:              && player.get_y() == monster.get_y());
58:  }
59:
60:  /*
61:      Move the player to a new location based on the user input. You
62:      to modify this if there are extra controls you want to add.
63:
64:      All key presses start with "TK_" then the character. So "TK_A"
65:      key being pressed.
66:
67:      A look-up table might be useful.
68:      You could decide to learn about switch statements and use them h
69:  */
70:
71:  void movePlayer(int key, Actor & player,
72:                  int game_world[NUM_BOARD_X][NUM_BOARD_Y])
73:  {
74:      int yMove = 0, xMove = 0;
75:      if (key == TK_UP)
76:          yMove = -MAN_STEP;
77:      else if (key == TK_DOWN)
78:          yMove = MAN_STEP;
79:      else if (key == TK_LEFT)
80:          xMove = -MAN_STEP;
81:      else if (key == TK_RIGHT)
82:          xMove = MAN_STEP;
83:
84:      if (player.can_move(xMove, yMove)
85:          && game_world[player.get_x()+xMove][player.get_y()+yMove]
86:          != SHALL_NOT_PASS)
87:          player.update_location(xMove, yMove);
88:  }
89:
90:  void move_the_Goose(Actor &player, Actor &goose,
91:                       int game_world[NUM_BOARD_X][NUM_BOARD_Y])
92:  {
93:      int x_move=0;
94:      int y_move=0;
95:
96:      if(player.get_x()<goose.get_x())
97:      {
98:          x_move=-GOOSE_STEP;

```

```

99:     }
100:     else if(player.get_x()>goose.get_x())
101:     {
102:         x_move=GOOSE_STEP;
103:     }
104:
105:     if(player.get_y()<goose.get_y())
106:     {
107:         y_move=-GOOSE_STEP;
108:     }
109:     else if(player.get_y()>goose.get_y())
110:     {
111:         y_move=GOOSE_STEP;
112:     }
113:     if (goose.can_move(x_move, y_move))
114:         goose.update_location(x_move, y_move);
115:
116: }
117: /*
118:      What other functions do you need to make the game work? What can
119:      add to the basic functionality to make it more fun to play?
120: */

```