

```

/*
gooseEscapeMain.cpp
Jaden Durnford and Dennis Li
*/
#include <BearLibTerminal.h>
#include <cmath>
#include <ctime>
#include <iostream> // Debugging tip: You can still use cout to put debug
messages on the regular console window
using namespace std;
#include "gooseEscapeUtil.hpp"
#include "gooseEscapeActor.hpp"
#include "gooseEscapeConsole.hpp"
#include "gooseEscapeGamePlay.hpp"
#include <iostream>
#include <fstream>

//set up the console. Don't modify this line!
Console out;

void readFile(int gameBoard[20][70])
{
    ifstream fin("gameBoard.txt");
    for (int y = 0; y<20;y++)
    {
        for (int x = 0;x<70;x++)
            fin >> gameBoard[y][x];
        cout << endl;
    }

    fin.close();
}

int main()
{
    // Set up for using rand() to generate random numbers (can be deleted if random
numbers not used)
    srand(time(NULL));

    //Set up the window. Don't edit these two lines
    terminal_open();
    terminal_set(SETUP_MESSAGE);

/*
    The code below provides a skeleton of the game play. You will need to
    write code for setting up the game board, and playing the game itself.
    You can modify the code given as needed.

    Call the functions that you have written in the game play file.
*/
}

```

```

// declare the game board "map"
int gameBoard[20][70] = {0};

/*
    Initialize locations in the game board to have game features. This
    would include anything that is static and doesn't move like a wall. Hard
    coding them like you see below is a poor way to code this. What if you
    have many features to add to the game board? Should you use a loop? Does
    it make sense to store this information in a file? Should code be a
    function as well?
*/
readFile(gameBoard);

// Call the function to print the game board
printBoard(gameBoard);
/*
    The player and goose are initialized to the same locations
    (10,10) and (70,20) each time, with health of 100.
    You likely want to change this somehow.
*/
// make the player
Actor player(PLAYER_CHAR, 20,2, 100, PLAYER_COLOUR);

// make the goose
Actor monster(MONSTER_CHAR, 68,18, 100, GOOSE_COLOUR);

// Printing the instructions in the console window
out.writeLine("Escape the Goose! " + monster.get_location_string());
out.writeLine("Use the arrow keys to move");
out.writeLine("If the goose catches you, you lose!");
out.writeLine("Be careful! Sometimes the goose can jump through walls!");

/*
    This is the main game loop. It continues to let the player give input
    as long as they do not press escape or close, they are not captured by
    the goose, and they didn't reach the win tile
*/
/*
    All key presses start with "TK_" then the character. So "TK_A" is the "a"
    key being pressed.
*/
int keyEntered = TK_A; // can be any valid value that is not ESCAPE or CLOSE
int moveCounter = 0;
bool gameRunning = true;

while(keyEntered != TK_ESCAPE && keyEntered != TK_CLOSE
      && !captured(player,monster) && gameRunning)
{
    // get player key press
    keyEntered = terminal_read();

```

```

    if (keyEntered != TK_ESCAPE && keyEntered != TK_CLOSE)
    {
        // move the player, you can modify this function
        movePlayer(keyEntered,player,gameBoard);

        // call the goose's chase function, with an 80% chance that the goose
gets to move when the player makes a move
        moveCounter = rand() % 100;
        if (moveCounter < 19)
        {
            moveGoose(player,monster,gameBoard);
        }

        // call other functions to do stuff?
    }

    if (gameBoard[player.get_y()][player.get_x()] == WINNER)
        gameRunning = false;
}

if (keyEntered != TK_CLOSE)
{
    //once we're out of the loop, the game is over
    out.writeLine("Game has ended");

    // Output why:  did the goose get you?  Or did you win?

    // Wait until user closes the window
    while (terminal_read() != TK_CLOSE);
}

//game is done, close it
terminal_close();
}

```