
```
/**GOOSE ESCAPE GAME PLAY HEADER FILE BEGINNING!!***/
```

```
#ifndef GOOSE_ESCAPE_GAMEPLAY
#define GOOSE_ESCAPE_GAMEPLAY
#include "gooseEscapeUtil.hpp"
#include "gooseEscapeActors.hpp"
#include "gooseEscapeConsole.hpp"

/*This file is all about the game world. You will modify this to add
  constants and function prototypes. Modify gooseGamePlay.cpp to
  actually add functionality.
*/

/*
  Declare constants to indicate various game world features in the board
  array. Modify them to fit what you would like to do in the game. You can
  change the type if you choose to store your game board as something other
  than integers.
*/
// Going further: Learn how to use an enum for these values
const int EMPTY = 0;
const int SHALL_NOT_PASS = 1;
const int WINNER = 2;
const int TELEPORT = 3;

/*
  A few examples of characters both for actors and for the game board
  itself are shown.
*/
//display characters
const int PLAYER_CHAR = int('@');
const int MONSTER_CHAR = int('G');
const int WALL_CHAR = int('#');
const int WIN_CHAR = int('%'); // % sign, a special character used in the ancient game ↗
    "Rogue"
const int BLANK_CHAR = int('-');
const int TELEPORT_CHAR = int('*');

/*
  Game play function prototypes are give below
*/

// print the game board function prototype
void printGameBoard(int gameBoard[NUM_BOARD_Y][NUM_BOARD_X]);

//Updates the private static variables to use inside the function
void sendGameBoardCoordinates(int teleX1, int teleY1, int teleX2, int teleY2);
```

```
/*
    Do something when the goose captures the player

    If you want to attack or something else, this is the function you
    need to change. For example, maybe the two touch each other and
    then fight. You could add a health to the Actor class that is
    updated. Run, use weapons, it's up to you!
*/
bool captured(Actor const & player, Actor const & monster);

/*
    Move the player to a new location based on the user input. You may want
    to modify this if there are extra controls you want to add.

    All key presses start with "TK_" then the character. So "TK_A" is the a
    key being pressed.

    A look-up table might be useful.
    You could decide to learn about switch statements and use them here.
*/

void movePlayer(int key, Actor & player, int gameBoard[NUM_SCREEN_Y][NUM_SCREEN_X]);

/*
    What other functions do you need to make the game work? What can you
    add to the basic functionality to make it more fun to play?
*/

//Function that returns if the player has won or not
bool won(Actor &player, int gameBoard[NUM_SCREEN_Y][NUM_SCREEN_X]);

//Confirms if the player can teleport
bool isTeleportable(int gameBoard[NUM_BOARD_Y][NUM_BOARD_X], Actor &player);

//teleports the actor. Ensure that isTeleportable() is called beforehand!
void teleportActor(Actor &player);

//Called to refresh the position of the teleporters!
void place_teleporter();

//Updates and refreshes player location
void updatePlayerLocation(Actor &player, int xMove, int yMove);

//Chases the player
void chasePlayer(Actor & monster, Actor const & player, int gameBoard[NUM_BOARD_Y]
    [NUM_BOARD_X]);

#endif
```