

Homework 3

Solution:

1. Please send solution to: zbaharav@cogswell.edu
2. You know the drill by now: Simply hit reply, and no zipped directories etc.. Just ascii-files or Word documents (or equivalent)

====

1. Dangeon game:
 - a. Beautify the printout to include 'coordinates' and 'cells'.
 - i. Then, implement it using the overloading of the "<<" operator.
 - b. Implement a 'movePlayer' method. The input from the user is 'u', 'd', 'r', 'l'.
 - i. Make sure you cannot fall overboard...
 - c. Define two new variables, 'lives' and 'cash', that will hold the players status:
 - i. Lives decreases every time you fall into a trap.
 - ii. Cash increases anytime you hit the gold.
 - d. When printing the board, also print the players cash and live status.
 - e. Make some game logic. For example: Zero lives==> you lose. Reaching the Exit: Game is done. Move counter, etc.
 - f. If you really want, you can add enemies that pop-up randomly, and act like traps..

Showing just the printing style:

```

C:\Users\Zach\Dropbox\z\Cpp\Teach_2014\Dungeon1\Debug\Dungeon1.exe
Starting program?
Printing Board: Non fancy style.
.....
.P.....
.t.....
.g.....
.....
.....G..
.....I...
.....X
Cash: 0 ; Lives: 5

Printing Board: Fancy style.
 0 1 2 3 4 5 6 7 8 9
0 | . | . | . | . | . | . | . | . | . |
1 | . | P | . | . | . | . | . | . | . | . |
2 | . | . | . | . | . | t | . | . | . | . | . |
3 | . | . | . | g | . | . | . | . | . | . | . |
4 | . | . | . | . | . | . | . | . | . | . | . |
5 | . | . | . | . | . | . | . | . | . | . | . |
6 | . | . | . | . | . | . | . | . | G | . | . | . |
7 | . | . | . | . | . | . | . | T | . | . | . | . |
8 | . | . | . | . | . | . | . | . | . | . | . | . |
9 | . | . | . | . | . | . | . | . | . | . | . | X |
Cash: 0 ; Lives: 5
Enter command (h for help, Q to quit):
  
```

And the moving part (but not collecting prizes, or losing lives):



```
C:\Users\Zachi\Dropbox\zCcpp\Teach_2014\Dungeon1\Debug\Dungeon1.exe
Printing Board: Non fancy style.
.....
.P.....
...E.....
..G.....
.....
.....G..
.....T.....
.....X
.....X
Cash: 0 ; Lives: 5
Enter command (h for help, Q to quit): d
Printing Board: Non fancy style.
.....
.P.....
..E.....
..G.....
.....
.....G..
.....T.....
.....X
.....X
Cash: 0 ; Lives: 5
Enter command (h for help, Q to quit): r
Printing Board: Non fancy style.
.....
.P.t.....
..G.....
.....
.....G..
.....T.....
.....X
.....X
Cash: 0 ; Lives: 5
Enter command (h for help, Q to quit): r
Printing Board: Non fancy style.
.....
.Pt.....
..G.....
.....
.....G..
.....T.....
.....X
.....X
Cash: 0 ; Lives: 5
Enter command (h for help, Q to quit):
```

=== End of Homework 3 ===

// File main.cpp

```
#include <iostream>

#include "DGame.h"

using std::cout;
using std::cin;

int main(void)
{
    cout << "Starting program!\n";

    DGame dg;

    dg.printBoard();

    cout << dg;
```

```

char userInput=' ';

while (userInput != 'q')
{
    cout << "Enter command (h for help, Q to quit) : ";
    cin >> userInput;
    if (userInput < 'a') userInput+= 'a'-'A';
    switch (userInput)
    {
        case 'q':
            break;
        case 'h':
            cout << "Help? what help?!? You are on your own here...\n" ;
            break;
        case 'u':
        case 'd':
        case 'r':
        case 'l':
            dg.movePlayer(userInput);
            break;
    }
    dg.printBoard();
}

cout << "End of program!\n";

system("pause");

return 0;
}

```

// File DGame.h

```

#ifndef _DGAME_H
#define _DGAME_H

#include <iostream>

// size of Board
const int ROWS = 10;
const int COLS = 10;

enum BoardElements { bempty=0, bplayer1, bplayer2, btrap1, btrap2, bgold1, bgold2,
bexit};

class DGame
{

private:
    BoardElements Board[ROWS][COLS];
    int playersNum;           // one or two players
    int r1,c1, r2,c2;       // Coordinates of players 1 and 2.
    int cash;
    int lives;

```

```

        char BoardElementsToChars(BoardElements b) const;

public:
    DGame(int playersNumber = 1);
    void printBoard();
    int movePlayer(char command, int playerNum=1);

    friend std::ostream & operator <<(std::ostream & os, const DGame &dg);
};

#endif

```

// File DGame.cpp

```

#include "DGame.h"

// These would NOT work in the class declaration!
const char BoardElementsChars[] = {'.', 'P', 'Q', 't', 'T', 'g', 'G', 'X'};

// Constructor
DGame::DGame(int playersNumber)
{
    playersNum = playersNumber;

    // Initialize the board with Traps, treasures, etc
    for (int rr=0; rr<ROWS; ++rr)
        for (int cc=0; cc<COLS; ++cc)
        {
            Board[rr][cc] = bempty ;
        }

    r1 = 1, c1 = 1;
    Board[r1][c1] = bplayer1 ;

    Board[2][4] = btrap1 ;
    Board[3][2] = bgold1 ;
    Board[7][5] = btrap2 ;
    Board[6][7] = bgold2 ;

    Board[ROWS-1][COLS-1] = bexit ;

    //
    cash = 0;
    lives = 5;
}

void DGame::printBoard()
{
    std::cout << "\nPrinting Board: Non fancy style.\n";
    for (int rr=0; rr<ROWS; ++rr)
    {
        for (int cc=0; cc<COLS; ++cc)
        {

```

```

        //std::cout << Board[rr][cc] ;
        //char c = BoardElementsChars[(int) Board[rr][cc]]; //
Promoting enum to int.
        char c = BoardElementsToChars(Board[rr][cc]);
        std::cout << c;
    }
    std::cout << std::endl;
}
std::cout << "Cash: " << cash << " ; Lives: " << lives << std::endl;
}

int DGame::movePlayer(char command, int playerNum)
{
    Board[r1][c1] = bempty ;
    switch (command)
    {
    case 'u':
        if (r1) r1--;
        break;
    case 'd':
        if (r1<(ROWS-1)) r1++;
        break;
    case 'r':
        if (c1<(COLS-1)) c1++;
        break;
    case 'l':
        if (c1) c1--;
        break;
    }

    Board[r1][c1] = bplayer1 ;

    return playerNum;
}

std::ostream & operator<<(std::ostream & os, const DGame & dg)
{
    std::cout << "\n\nPrinting Board: Fancy style.\n";

    std::cout << " " << " " << " " ;
    for (int cc=0; cc<COLS; ++cc)
        std::cout << " " << cc <<" " ;
    std::cout << std::endl;

    std::cout << " " ;
    for (int cc=0; cc<COLS; ++cc)
        std::cout << "----" ;
    std::cout << std::endl;

    for (int rr=0; rr<ROWS; ++rr)
    {
        std::cout << " " << rr << " | " ;
        for (int cc=0; cc<COLS; ++cc)
        {
            //char c = BoardElementsChars[(int) dg.Board[rr][cc]]; //
Promoting enum to int.

```

```

        char c = dg.BoardElementsToChars(dg.Board[rr][cc]);
        std::cout << c << " | ";
    }
    std::cout << std::endl;
    std::cout << "    " ;
    for (int cc=0; cc<COLS; ++cc)
        std::cout << "----" ;
    std::cout << std::endl;
}

std::cout << "Cash: " << dg.cash << " ; Lives: " << dg.lives << std::endl;

return os;
}

char DGame::BoardElementsToChars(BoardElements b) const
{
    char c;
    switch(b)
    {
    case bempty:
        c='.';
        break;
    case bplayer1:
        c='P';
        break;
    case btrap1:
        c='t';
        break;
    case btrap2:
        c='T';
        break;
    case bgold1:
        c='g';
        break;
    case bgold2:
        c='G';
        break;
    case bexit:
        c='X';
        break;
    default:
        c='?';
    }
    return c;
}

```

==== End of file =====