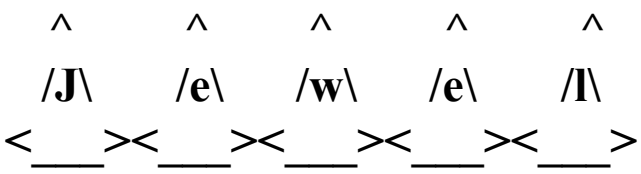
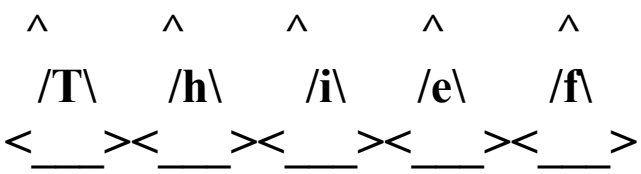




Part II

By Tyler Stevens

CSC-5-48102

Dr. Mark Lehr

December 12, 2016

Table of Contents

Introduction

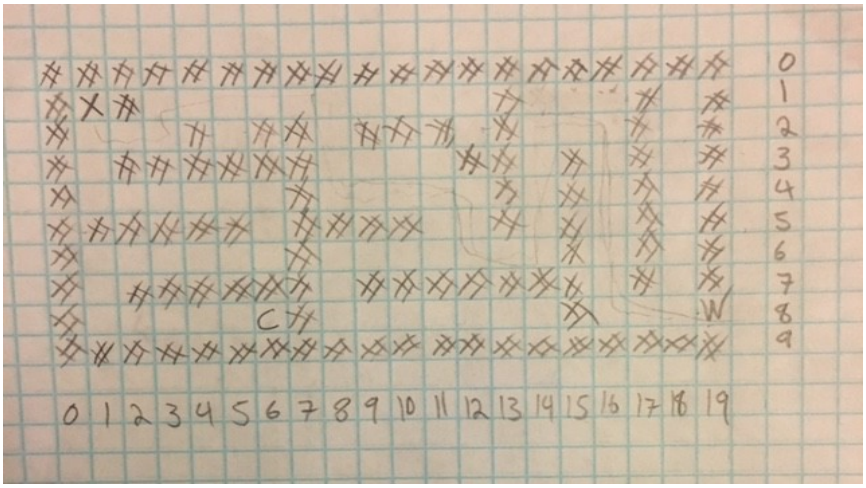
Title: Jewel Thief

This is the second installment in the Jewel Thief series and follows a skilled thief that has been hired by an unnamed man to steal precious jewels from various places. One of his targets is to steal the “Regent” diamond from The Lovre museum in Paris. Once the rare item is secure in the thieves possession all there is to do now is escape the maze like museum without getting caught. Escaping is harder than the thief anticipated. Dead ends and super natural beings litter the museum. Are you brave enough to be the Jewel Thief?

Summary

Project size: 377 lines of code.

The meat of this program is in the actual maze system. In the previous game it was created with a 2D array so I stuck with its basic mechanic for the sequel. The way I think of it is it works very much like a graph where you have the Y axis and the X axis. That being said I just plotted points in the array like a graph. First, I started out with the width of the maze at 20 and its height at 10. After that I drew the maze out spaces and pound signs. Jewel Thief part one was lacking multiple mazes so I got to implement 2 more in its predecessor. To do this I had a switch statement that according to the users input would copy a certain 2D array to the one that the meat of the program uses. Before I really understood 2D arrays the tricky part to figure out was implementing character movement in the array, but now nothing could be simpler. The Thief starts at position 1 on the X axis and also on the Y axis. His position is manipulated by increasing and decreasing his value in the maze with 4 separate for loops that increment either Y position or the X position. This is achieved by inputting 'a' for left, 'd' for right, 'w' for up, and 's' for down. It is all contained in a switch statement which also includes a quit option and a instruction prompt. The switch is contained in a do while loop that runs as long as the variable quit is set to false. Once the thief is at a certain value in the array and if statement sets quit to true ending the do while loop. Setting quit is also how the player can exit the program by entering 'q'. When 'q' is entered quit is set to true in the switch statement. Part II also include a file system that saves previous players names and uses a selection sort to display them from Z to A.



- Original blueprints

Description

The whole point of this game is to solve the maze and escape! Hint: go to space 6, 3 in the maze for a cheat code.

Pseudo code

Input/Output
File objects
String objects

Global Constants
Hold 20 names in a file

Function Prototypes
Display title from a file
Copy names from file to one dimensional array
Sort Previous players names in Z to A order
Display sorted array

int main(int argc, char** argv) {

Declaration of Variables and Process values

Width and Height of Maze
User input. “asdw” to move “q” to quit, and “i” for instructions.
Choose a maze 1, 2, or 3
2D Array Maze
Record players names

Input Values
Default thief position position x and position y
If false execute program is true quit program.
Symbol for thief is “x”

2D Array Mazes
First maze
Second maze
Third Maze

Call Function
Display title from a file

Instructions

Choose maze
Input validation. Input must 1,2,or 3
case 1 copies maze 1
case 2 copes maze 2
case 3 copies maze 3

Do while quit does not equal true execute code
Placement of Thief in maze. Default position is maze[1][1]
case “a” move to the left
case “d” move to the right
case “w” move up
case “s” move down
case “q” set quit to true and exit program
case “r” resets value of thief to original position

Winning criteria
Solving the maze exits the program when position x equals 9 and position y equals 19

Cheat code
If user inputs the secret code the automatically win from position 6,3.

Exits program

Title Screen Function
Purpose: Show rule of 72
Inputs: Inputs to the function here -> Description, Range, Units
Output: Outputs to the function here -> Description, Range, Units
title screen that is saved in the file "title.txt

Declaration of Variables
Any input will execute program
Variable for the title
Declare input file object
Open input file object "title.txt"

If file was opened successfully
While we can read in more input from the file
Display 'words'

Else
Error with file

Plays game when user enters anything from the keyboard

Exit Function

Selection Function
Purpose: Sort contents of array from Z to A order
Inputs: Inputs to the function here -> Description, Range, Units
array[]-> 1 dimensional array of strings
size->size of array, 20
Output: Outputs to the function here -> Description, Range, Units
Sorted array

Declare Variables
Minimum index
Minimum Value

Selection Sort

Show Array
Purpose: Display contents of array
Inputs: Inputs to the function here -> Description, Range, Units
array[]-> 1 dimensional array of strings
size->size of array, 20
Output: Outputs to the function here -> Description, Range, Units
Players names in z to A order

Display output
Step through array with a for loop to display each name

Exit Function

Player names Function
Purpose: Copy contents of a file to a 1 dimensional array
Inputs: Inputs to the function here -> Description, Range, Units
array[]->1 dimensional array of strings
size->size of array, 20
Output: Outputs to the function here ->
Display names in array

Declaration of Variables
copy players name to file
copy from a file

Update file
Open the file in append mode.
Writing more data to file
Close file
Open file
Copy file to array
Close file

Sort Array
Call selection sort function

Display values again
Call show array function

Variables

Main
const char **W = 20, H = 10;** //Width and Height of Maze
unsigned char **thief = 'X' ;** //symbol for thief position
unsigned char **action;** //user input. asdw to move and q to quit
int **posx = 1, posy = 1**; //default player
bool **quit=false;**
unsigned char **maze[H][W] , maze1[H][W] , maze2[H][W] , maze3[H][W] ,**
string **name;**
unsigned short **choose;**
int **prevposx = posx;**
int **prevposy = posy;**
unsigned char **space = {32};**

void titleScreen
char **play**; //any input will execute program
string **title**; //variable for the title
ifstream **inFile;** //declare input file object

Void selectionSort
Int **=minIndex;**
string **mine=Value;**
int **startscan=0;**

Void plaerNames
ofstream **dataFile;**
ifstream **outFile;**
string **thiefs[SIZE];**

Code

```
/*
 * File:   main.cpp
 * Author: Tyler
 * Created on October 25, 2016, 9:10 PM
 * Purpose: Create a maze game the user figures out how to escape
 */

#include <iostream> //Input/Output
#include <cstdlib>
#include <fstream> //File objects
#include <string> //String objects
using namespace std;
//User Libraries

//Global Constants

//Execution Begins Here!

 //Function prototypes
void titleScreen();

//Execution Begins Here!
    int main() {
    //Declare Variables
    const char W = 20, H = 10;//Width and Height of Maze
    unsigned char thief = 'X';//symbol for thief position
    unsigned char action;//user input. asdw to move and q to quit
    int posx = 1, posy = 1;//default player
    int choose;
    bool quit=false;

    //2D Array Maze
   unsigned char maze[H][W] = {//second maze
        {'#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'}, //0
        {'#', ' ', '#', ' ', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', '#', ' ', '#', ' ', ' ', ' ', '#'}, //1
        {'#', ' ', '#', '#', '#', ' ', '#', '#', ' ', '#', ' ', '#', ' ', '#', ' ', '#', ' ', '#', '#', '#'}, //2
        {'#', ' ', ' ', ' ', '#', ' ', ' ', '#', ' ', ' ', ' ', '#', ' ', ' ', ' ', '#', ' ', '#', ' ', '#'}, //3
        {'#', ' ', '#', ' ', '#', ' ', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', ' ', '#', ' ', '#'}, //4
        {'#', ' ', '#', '#', '#', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '#', ' ', '#', ' ', '#'}, //5
        {'#', ' ', ' ', '#', ' ', ' ', '#', '#', ' ', '#', '#', '#', ' ', '#', ' ', ' ', ' ', '#', ' ', '#'}, //6
        {'#', ' ', '#', '#', ' ', '#', '#', ' ', ' ', '#', ' ', '#', ' ', '#', '#', '#', ' ', ' ', ' ', '#'}, //7
        {'#', ' ', ' ', ' ', ' ', '#', ' ', ' ', '#', '#', ' ', ' ', ' ', ' ', '#', ' ', ' ', '#', ' ', ' '}, //8
        {'#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'},
    }; //9
    //0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
```

```
//Call Function
titleScreen();

cout<<"You have got the jewel. Now Escape!\n\n";

//Instruction
cout<<"INSTRUCTIONS \n";
cout<<"-------------------------\n";
cout<<"Move Thief Left with:  'a' \n";
cout<<"Move Thief Right with: 'd' \n";
cout<<"Move Thief Up with:    'w' \n";
cout<<"Move Thief Down with:  's' \n";
cout<<"Quit game with:        'q' \n";
cout<<"See instructions with: 'i' \n";
cout<<"-------------------------\n";
//Quits game when user enter q
 do{

   //Placement of Thief in maze. Default position is maze[1][1]
   maze[posx][posy] = thief;

  //Increments 2D array
   for (int y = 0; y < H; y++) {
      cout << endl;

      for (int x = 0; x < W; x++){
         cout << maze[y][x];
      }
   }

   //Inputs direction the thief moves by
   cout << "\nAction:";
   cin>>action;

   //Declaration of variables
   int prevposx = posx;
   int prevposy = posy;
   unsigned char space = {32};

   //Input values for Thief movement
   switch (action){
      //Moves thief to the left in the maze
      case'a':
         if (maze[posx][posy - 1] != '#'){
            posy--;
```

```
        cout<<posx<<','<<posy<<endl;//displays move
        maze[prevposx] [prevposy] = space;
    } //replace 'x' with space
    system("clear");//clear screen
    break;

//Moves thief to the right in the maze
case'd':
    if (maze[posx][posy + 1] != '#') {
        posy++;
        cout<<posx<<','<<posy<<endl;
        maze[prevposx] [prevposy] = space;
    }
    system("clear");
    break;

//Moves thief down in the maze
case's':
     if (maze[posx + 1][posy] != '#'){
        posx++;
        cout<<posx<<','<<posy<<endl;
        maze[prevposx] [prevposy] = space;
     }
    system("clear");
    break;

//Moves thief down in the maze
case'w':
    if (maze[posx - 1][posy] != '#'){
        posx--;
        cout<<posx<<','<<posy<<endl;
        maze[prevposx] [prevposy] = space;
    }
    system("clear");
    break;

//Display instruction on how to play the game
case'i':
    cout<<"INSTRUCTIONS \n";
    cout<<"--------------------------\n";
    cout<<"Move Thief Left with:  'a' \n";
    cout<<"Move Thief Right with: 'd' \n";
    cout<<"Move Thief Up with:    'w' \n";
    cout<<"Move Thief Down with:  's' \n";
    cout<<"Quit game with:        'q' \n";
    cout<<"See instructions with: 'i' \n";
```

```
            cout<<"--------------------------\n";
        break;

        //Quits game
        case'q':
            quit=true;
            cout<<"See you later!\n\n";
            break;

        //If user does not input asdwq they will be prompt the instruction again
        default:
            cout << "Use asdw to move! hurry before you're caught!";
    }

    //Solving the maze exits the program
    if((posx == 8) && (posy == 19 )){
        cout<<"You escaped!"<<endl;
        quit=true;
    }

    //A cheat code a Genie whispers to the thief
    if((posx == 3) && (posy == 6 )){
        cout<<"A magic entity rises from a lamp and whispers to you \n"
            "Enter assddddddddsddsddsdd at once"<<endl;
    }
  }while(quit != true);

  return 0;
}
void titleScreen(){
  //Declaration of Variables
  char play;//any input will execute program
  string title;//variable for the title
  ifstream inFile;  //declare input file object
  inFile.open("files.txt"); //open input file object "file.txt"

  if (inFile) { //if inFile was opened successfully
    while (inFile >> title) {
      getline(inFile,title);//while we can read in more input from the file
      cout << title << endl; //display 'words'
    }
   //Unsuccessfully tries to open file
  }else {
    cout << "Error with file" << endl;
  }
```

```
  //User input starts game
   cin>>play;

   return;
}
```

Flow Chart

https://www.gliffy.com/go/share/sawu95l3wkr37075akn4