

Rhys Lewis

by Rhys Thomas Lewis

FILE	12025_RHYS_THOMAS_LEWIS_RHYS_LEWIS_311314_938693423.DOCX (409.61K)		
TIME SUBMITTED	11-JAN-2021 02:17PM (UTC+0000)	WORD COUNT	1403
SUBMISSION ID	141595457	CHARACTER COUNT	6411

Graphics and Games Programming

3D OGL GAMES PROGRAMMING

RHYS THOMAS LEWIS (1802146)

Table of Contents

Program Structure	2
Algorithms.....	3
Loading OBJ Files.....	3
Random Placement.....	3
Giving Positions and Rendering Objects	3
Cube Selection	3
Data Structures	5
Classes.....	5
Structs	5
User Guide	6
Test Schedule	7
Screenshots.....	10

Program Structure

The program functions like so: -

- On bootup, the program randomly places pairs of models of the numbers 1-5 in a number pad orientation.
- The program covers the numbers with cube objects
- The problem displays a lit display with the ordered numbers hidden by cubes
- The user presses numbers 1-9 on their keypad
- The corresponding number cube is removed, revealing the number underneath
- Once the player selects 2 cubes, if the numbers underneath are the same, they are not re-covered. If they are not, they are covered up when the player picks another cube.
- Once every cube has been permanently revealed, it outputs a victory message and tells you how many times you guessed wrong throughout the game
- The game restarts automatically with new random placements.

The program is a combination of both unique code written by the author and assets supplied by the university via their online resources.

Figure 3: Selection Function

timer or immediately. The system fills the entry with the number object given. On the next Numpad press, it checks if that entry is empty and that the current keypress is not the same as the last, if it is, it sets that entry to the new number object and compares it to the first. If both entries are the same, the system sets it to “matched”, so it isn’t hidden again. If not, it increments the wrong guess count. It then empties the array for use on the next keypress.

Data Structures

Classes

Both the OBJ loaded objects and the cubes that cover them up are Classes, this allows the author to structure the objects to include both variables and functions, making for adaptable objects. For instance, the class "cCube" includes both an OBJFile object, containing the resources needed to render the cube, Boolean variables to check if its matched, and a pointer to the OBJFile type NumberObject underneath, so it is easy to access the identity of the number is being hidden underneath. A class is also used to hold the groups of numbers, as the fact that it's a class allows the author to install functions like "RandomisePlacements()" into the model object, making it easier to use.

Structs

Structs are used very sparingly, being used only to define data types (vector, polygon, etc.) for use in the objLoader.

User Guide

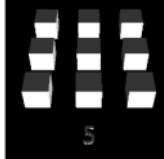
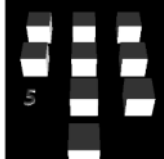
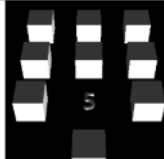
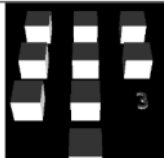
Here is a guide on how to play the game:

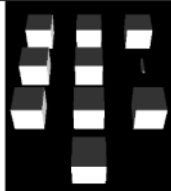
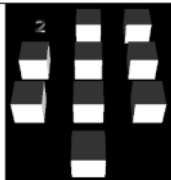
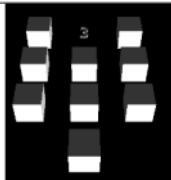
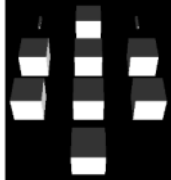
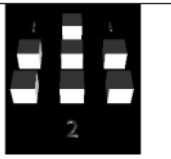
1. Open and boot the program.
2. Press a number on your keypad to reveal a number hidden by a cube, the numbers are formatted to a standard keyboard Numpad, or, if you do not have one, here is a reference.
3. Once the number has been another, select another number with the same method.
4. If the two numbers match, they will remain revealed as you select more numbers, if not, they will be hidden as you select another
5. Once you have matched all the numbers, the game will start again automatically, check the console window to see how many wrong guesses you had during the game!

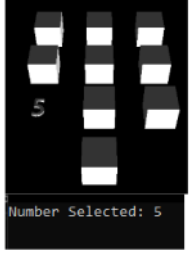
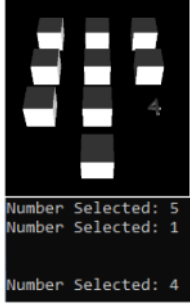
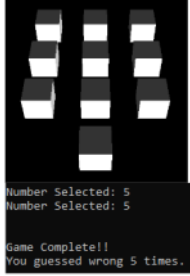
7	8	9
4	5	6
1	2	3
	0	

Test Schedule

To make sure the program works as planned, vigorous tests will be conducted, here is a record of those tests.

Test Number	Test Goal	Prerequisites	Expected Results	Pass/Fail
1	The program renders 9 cubes on boot	•	The program displays 9 cubes in proper order	 Pass
2	Pressing the 0 key reveals the bottom-most cube	• Program is running	the bottom-most cube will be revealed.	 Pass
3	Pressing the 1 key reveals the cube in the (3, 1) position	• Program is running	(3, 1) cube will be revealed.	 Pass
4	Pressing the 2 key reveals the cube in the (3, 2) position	• Program is running	(3, 2) cube will be revealed.	 Pass
5	Pressing the 3 key reveals the cube in the (3, 3) position	• Program is running	(3, 3) cube will be revealed.	 Pass
6	Pressing the 4 key reveals the cube in the (2, 1) position	• Program is running	(2, 1) cube will be revealed.	 Pass

7	Pressing the 5 key reveals the cube in the (2, 2) position	Program is running	(2, 2) cube will be revealed.	 Pass
8	Pressing the 6 key reveals the cube in the (2, 3) position	Program is running	(2, 3) cube will be revealed.	 Pass
9	Pressing the 7 key reveals the cube in the (1, 1) position	Program is running	(1, 1) cube will be revealed.	 Pass
10	Pressing the 8 key reveals the cube in the (1, 2) position	Program is running	(1, 2) cube will be revealed.	 Pass
11	Pressing the 9 key reveals the cube in the (1, 3) position	Program is running	(1, 3) cube will be revealed.	 Pass
12	Pressing a key after another will show reveal both cubes	- Program is running - A cube has already been revealed	Both cubes would be revealed	 Pass
13	Pressing a key after a matched pair will reveal an additional square while keeping the matched pair visible	- Program is running - A set of cubes have been matched	The new selection will be revealed while the matched pair stay revealed	 Pass

14	Selecting a cube outputs the revealed number to the command window	Program is running	The command window will output "Selected Number #", where # is the revealed number	 Pass
15	Pressing a key after an unmatched pair will reveal an additional square while re-covering the original	- Program is running - An unmatching pair is selected	A new box will be selected, previous 2 will be covered up	 Pass
16	After every number is paired, a congratulations message is displayed in the command window and the board is reset	- Program is Running - Every number is paired	Numbers will all be re-covered, the system window will output a victory message and a number of wrong guesses.	 Pass

Screenshots

```
void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 'Y':
            //Rotates camera
            theta = (theta + 10) % 360;
            glutPostRedisplay();
            break;
        case 'T':
            //Rotates camera
            theta = (theta - 10) % 360;
            glutPostRedisplay();
            break;
        case '0':
            Cubes[0].Reveal = true; //Sets relative cube to not render
            SelectionCheck(0);
            glutPostRedisplay();
            break;
        case '4':
            Cubes[1].Reveal = true; //Sets relative cube to not render
            SelectionCheck(1);
            glutPostRedisplay();
            break;
        case '1':
            Cubes[2].Reveal = true; //Sets relative cube to not render
```

Figure 6: Section of Keyboard function

```
#pragma once
#include "objloader.h"
#include <math.h>

class cCube
{
public:
    cCube();
    ~cCube();
    bool Reveal;
    bool Matched;
    OBJFile CubeOBJ;
    OBJFile *CoveredNumber;

private:
};

cCube::cCube()
{
    //Load Cube and set variables
    CubeOBJ.FileNamePoint = "cube.obj";
    Reveal = false;
    Matched = false;
    CubeOBJ.objloader(CubeOBJ.FileNamePoint);
}
```

Figure 4: Cube Class

```
#include "objloader.h"
#include <math.h>
#include <time.h>
#include <stdlib.h>

class Models
{
public:
    Models();
    ~Models();
    OBJFile *RandomPositions[10];
    void RandomisePlacements();

private:
    OBJFile *Num1Model;
    OBJFile *Num2Model;
    OBJFile *Num3Model;
    OBJFile *Num4Model;
    OBJFile *Num5Model;

    OBJFile *ModelList[10];
};

Models::Models()
{
    //Set Models for 5 numbers
    Num1Model = new (OBJFile);
    Num1Model.FileNamePoint = "1.obj";
```

Figure 5: Section of Models Class

FINAL GRADE

GENERAL COMMENTS

Instructor

71 /100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

GRADING FORM: SMU GENERIC_GJ

RHYS THOMAS LEWIS

71

RESEARCH-INFORMED LITERATURE



You've clearly research the literature and you do reference materials used in your report. Do use the school's standard ref style though. Essential topics covered here. No reference section apparent

5.5

KNOWLEDGE AND UNDERSTANDING



Some good work here demonstrating a good understanding of the requisite concepts and methods. Working OGL code shows your game demonstrates many of the concepts covered in class. Well done. Texturing missing but obj loaders implemented - good.

10

ANALYSIS



A good interpretation of the problem. A working solution synthesised and demonstrated.

18

PRACTICAL APPLICATION AND DEP.



Some good work here. Lighting and camera movement incorporated. Objects loaded from files. Good. No textures demonstrated. Shame. Selection via keyboard. This would be even better if you tried ray-casting for selection. Testing documented too - good.

35

SKILLS FOR PROFESSIONAL PRAC.



See comments earlier regarding referencing.
Testing documented. Report structured. Screen
shots helpful.

2.5