2D and 3D Graphics Engine (GE) Lab Requirements

HL

This lab counts total 10 points. Note IEEE style lab report has to be ready for the time to make on line submission, fail to submit an adequate IEEE style report can result in 5 marks reduction. This lab consists of 2 parts, the first part is for 2D GE and the second part is for 3D GE implementation.

To build 2D vector graphics processing engine, you will

1. Design and prototype LPC1769 micro-processor system board, and enable SPI LCD display interface.

2. Use draw line test code to run draw-a-line testing program. Once this is done, you are ready for the implementation of 2D screen saver functions 1 and 2. Function 1 is for rotating square patterns display and Function 2 is for generating trees.

3. Generate 2D screen saver of rotating squares based on vector graphics formula discussed in the class (5 points)

(1) use

$$P(x,y) = P\_1(x\_1,y\_1) + lamda * (P\_2(x\_2,y\_2) - P\_1(x\_1,y\_1))$$

with lamda = 0.8 by default, and lamda = 0.2 when prompted for user selected input;

(2) create two dimensional rotating patterns with data set of "parent" square;

(3) randomized location by using rand() function;

(4) randomized reduction of the parent square;

(5) choose one color for each set of rotation patterns, and rotates at least 10 levles or higher;

(6) continue to display each set of patterns without erasing the patterns.

4. Generate 2D trees with its branches level no less than 10 or higher based on vector graphics formula discussed in the class (5 points)

(1) use $P(x,y) = P\_1(x\_1,y\_1) + lamda * (P\_2(x\_2,y\_2) - P\_1(x\_1,y\_1))$ with lamda = 0.8 by default for tree branch reduction;

(2) create patch of forest by modifying one parent tree;

(3) randomized location of the new trees by using rand() function;

(4) randomized reduction of the parent tree trunks and branches;

(5) randomized angles for the branches;

(6) continue to display trees without erasing till the keyboard input detected.

5. The second part of this lab is 3D vector graphics engine, the computation requirements:

(1) Generate 3D cubes by designing your data points, and make it solid cube. The size of the cube is 100X100X100, and it is floating above x-y plane by 10. One back edge of the cube is overlapped with Zw axis as discussed in the example in the class; Define a point light source Ps(-50, 50, 200) and virtual camera E(200,200,200).

For the graphics computation,

(2) First display 3D coordinate system, with each axis as different color, e.g., r, g, b color;

(3) then display solid cube with S1, S2, and S3 surfaces;

(4) compute shadow with the Ps as shown in the lecture.

6. Submit project report together with

(1) exported project (source code is part of it) CANVAS subject to testing and verification.

7. Rubrics for the lab:

(1) satisfies the requirements stated above;

(2) lab report should cover (hardware part):

system block diagrams of the entire system setup including laptop computer;

system block diagram of the SPI colour LCD interface

Schematics of the LPC1769 interface to LCD colour display panel

table(s) of the pin connectivity

photo(s) of the implementation

(3) lab report (hardware side) should cover

Algorithm description

Flow chart(s)

Pseudo code

testing and verification section

source code listing (appendix)

8. Submit project report together with

(1) exported project (source code) subject to testing and verification, including compilation and build, as well as actual LPC1769 board testing

.

9. Rubrics for running program for the lab testing:

(1) Satisfies the requirements stated in 2018S-17-Lab-report-rubrics.txt (see githua/hualili/compe240 on line);

(2) lab report should cover (hardware side):

(2.1) system block diagrams of the entire system setup including laptop computer;

(2.2) system block diagram of the SPI color LCD interface;

(2.3) Schematics of the LPC1769 interface to LCD color display panel;

(2.4) table(s) of the pin connectivity;

(2.5) photo(s) of the implementation.

(3) lab report:

(3.1) software part should cover

a. Algorithm description;

b. Flow chart(s);

c. Pseudo code;

d. testing and verification section;

e. source code listing (appendix).

(END)