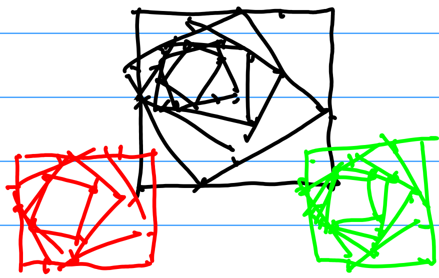
CMPE240

22

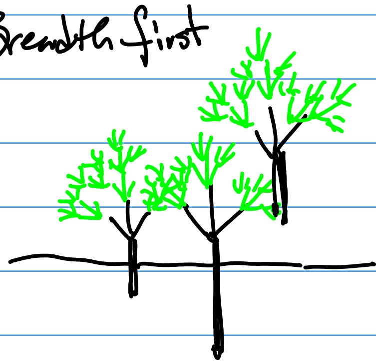
a. A set of Squares Rotation with one same color
b. Change location/color/size
c.



Keep the patterns, please don't erease them!

Part II Once Part I is Done Switch to Part I.

a. Breadth first

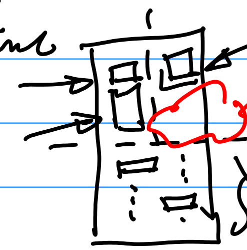


b. $L \geq 7$ c. Create A patch of Forest Changing Location/size

Note: Please Don't Erease the Drawing Till All the trees are Constructed.

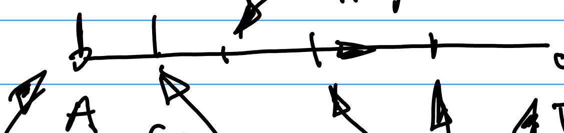
Note: Report Writting.

3D G.E. (Graphics Engine)

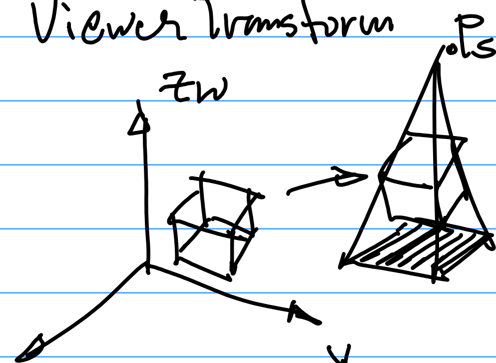


CPU Architecture
Jtag
CortexM3
mem. Contrl
3D G.E.
Peripheral Controllers.

GPP SPRs,
SPI SPRs.



A
B
Solid obj HL, HS. Removal
3D Concepts + Algorithm
(1) World-To-Viewer Transform
(2) Wireframe
Shade
State-of-the-Art
Lighting model. Diffuse Reflection

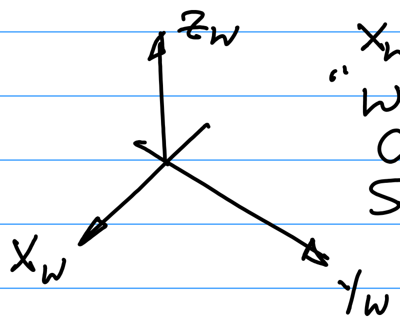


March 15 (monday)

Note: 1° midterm A week from this Wednesday (march 24th)
2° Smartphone (CAM Capability) A piece of Paper to the Class. Test the Environment

Ref: github: 2018F-114-3D Graphics

1°



X_w-Y_w-Z_w "World" Coordinate System.

Fig. 1

Right-Hand System,

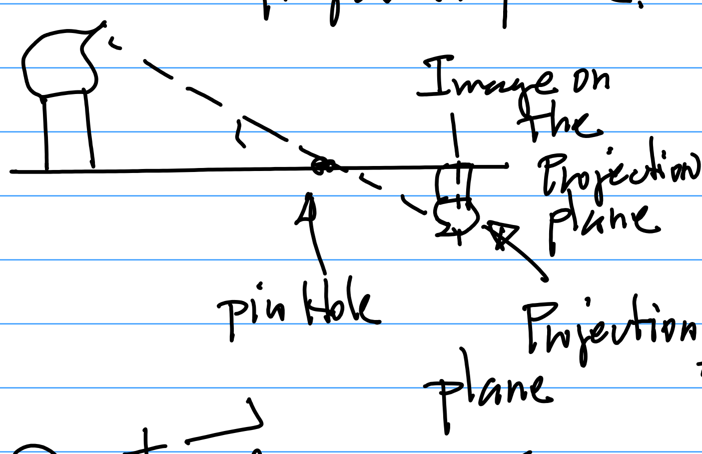
2° Virtual Camera

a. Virtual Enclosure

b. Optic Lens (Pin Hole)

$\Phi_d \simeq \phi$ Very Small diameter

c. Projection plane, the plane to form projected Image when light passing through the lens and Reaches the projection plane.



3. Denote $\vec{E}(x_e, y_e, z_e)$ as

Virtual Camera Location

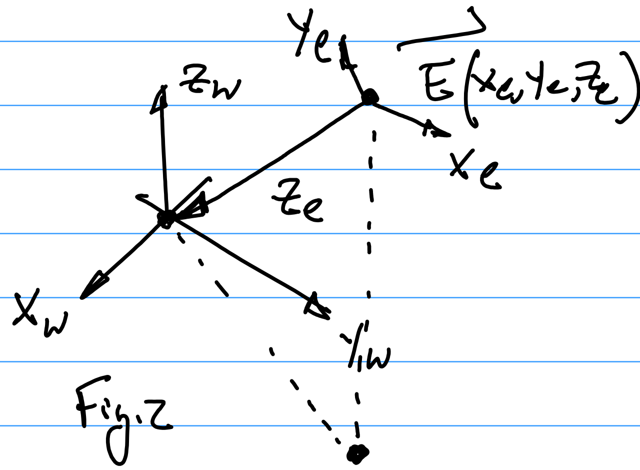


Fig. 2

Projection of $\vec{E}(x_e, y_e, z_e)$ onto X_w-Y_w plane in the X_w-Y_w-Z_w World Coordinate System

4. Viewer Coordinate system

Viewer ~ Virtual Camera

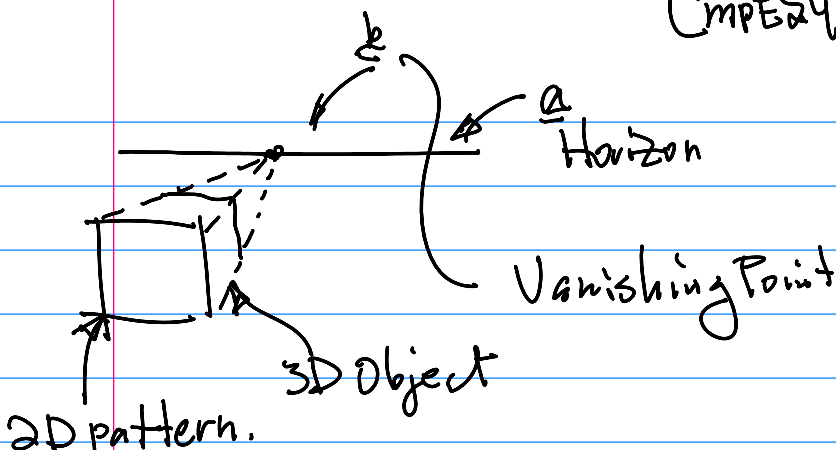
X_e-Y_e-Z_e Viewer Coordinate System. "Eye"

a. $\vec{E}(x_e, y_e, z_e)$ As the origin.

b. Left-Hand System.

c. Z_e-axis points to the origin.

5. Perspective Projection



3D Object formed on 2D plane

7. Transformation Pipeline

To Display 3D object(s) on to 2D LCD Display Screen,

- W2V (Work-2-Viewer) Transform
- Perspective Projection.

6. Perspective Projection → @ the Virtual Camera in X_w-Y_w-Z_w, in X_e-Y_e-Z_e.

{ Objects, $\vec{E}$ } in X_w-Y_w-Z_w → Viewer Coordinate System X_e-Y_e-Z_e → Projection Plane form Perspective Projection

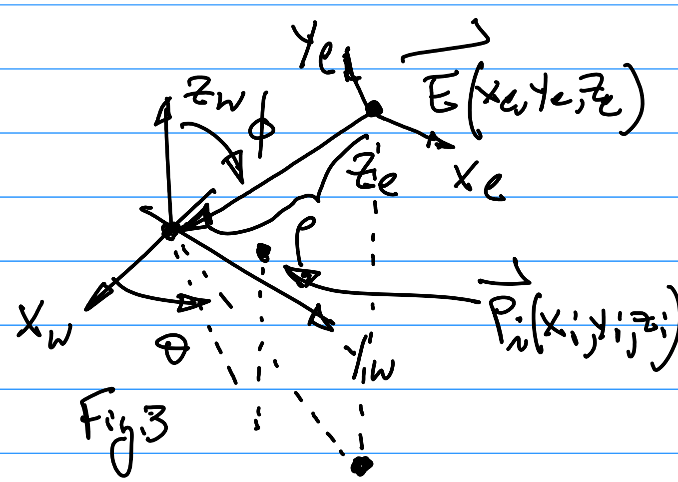


Fig.3

3 parameters:

a. theta θ Angle

b. Vector $\vec{E}$ to $\vec{O}$, " rol", Distance ρ

$\rho \triangleq \sqrt{x_e^2 + y_e^2 + z_e^2}$ ... (1)

c. Angle Z_w and Z_e ϕ.

Mathematical Formulation

- 1° Formula: Transformation Pipeline
  - a. World-to-Viewer Transform
  - b. Perspective Projection
- 2° Foundation
  - a. Translation
  - b. Rotations
    - ~ wrt X_w And
    - " Y_w wrt Arbitrary axis
    - " Z_w

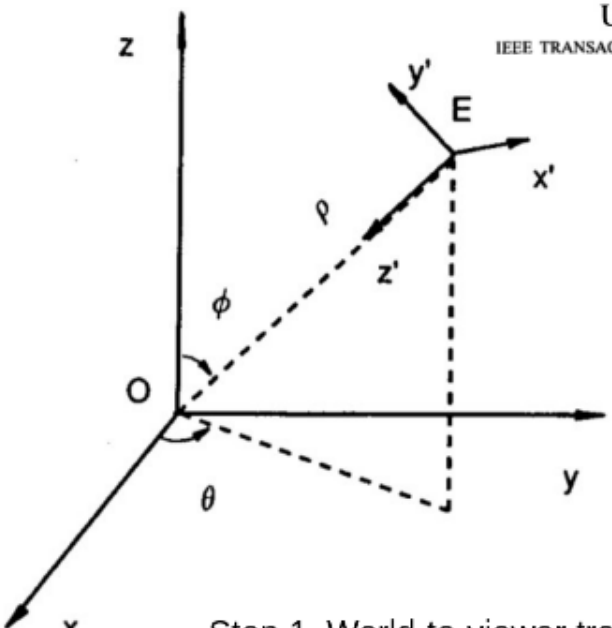
World-2-Viewer

Cmpe240

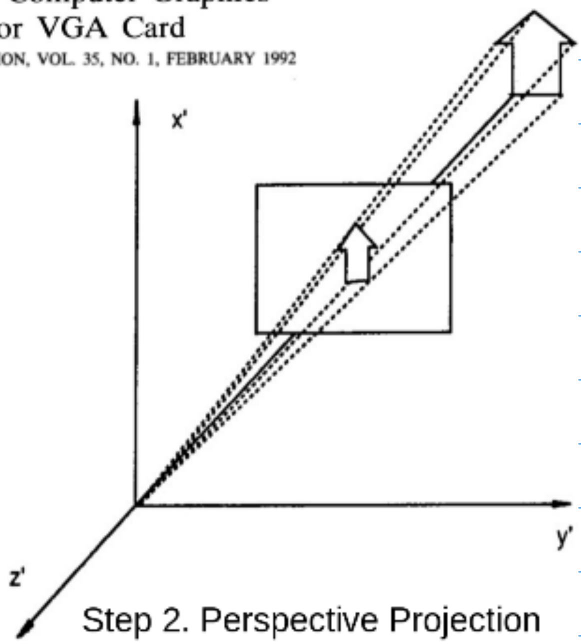
3D Transformation Pipeline Technique

Reference: H. Li Three-Dimensional Computer Graphics Using EGA or VGA Card

IEEE TRANSACTIONS ON EDUCATION, VOL. 35, NO. 1, FEBRUARY 1992



Step 1. World-to-viewer transform



Step 2. Perspective Projection

$$\mathbf{T} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi & 0 \\ -\sin\phi\cos\theta & -\sin\phi\cos\theta & -\cos\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_p = x_e\left(\frac{D}{z_e}\right)$$

$$y_p = y_e\left(\frac{D}{z_e}\right)$$

Harry Li, Ph.D.

CMPE240

Transform, map $\vec{P}_i$ from the World-coordinate to Viewer Coordinate.

$$\begin{pmatrix} x_i' \\ y_i' \\ z_i' \\ 1 \end{pmatrix} = T \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \quad \cdots (2)$$

After: X_e-Y_e-Z_e

Before: X_w-Y_w-Z_w

World to Viewer Transform.

$$\begin{pmatrix} \sin\theta & \cos\theta & 0 & 0 \\ \cos\theta & \sin\theta & " & " \\ \cos\theta & \cos\theta & " & " \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Part I. "θ"

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \cos\phi & \cos\phi & \sin\phi & " \\ \sin\phi & \sin\phi & \cos\phi & " \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Part II "ϕ"

$$\begin{pmatrix} - & & & \\ - & - & & \\ - & - & - & \rho \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Part III "_"

rho

March 17 (Wed)

Topics: 1° Hardware Architecture 2° Software SPRs Init & Config.

Example: 2D & 3D G.E.

SPI I/F LCD Display to Work with LPC1769.

GPIO (GPP) SPI

A set of 6 ~

- { System Configuration { Per. Cont., PWR, CLK
- * { Configuration of the Peripheral Cont. { Multiplexing

Compare to Rotation Matrix

$$\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

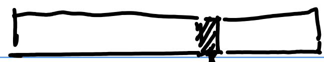
RTOS

Peripheral Controller { GPP, SPI

SPI's SPR.

1. Naming Convention
   LPC_SC → PCONP
2. Power Up the Selected Peripheral Controller By Setting the Corresponding.

CmpE240



"1"

Bitwise "OR"

$|= (1<<10)$

$\underbrace{\&=}_{\text{"AND"}} \sim(3<<20)$

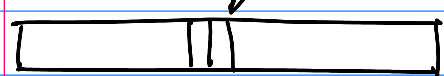
"11"

Negation, "00"

$\underbrace{|=}_{\text{"OR"}} (1<<20)$

"01"

Clear 2 Bits



Set 2 bits as "01"

$\underbrace{\&=}_{\text{}}\sim((3<<18)|(3<<16)|(3<<14));$

"AND" Nega. "11" Total 6 bits <<18

Clear 6 bits

$\underbrace{|=}_{\text{"OR"}} ((2<<18)|(2<<16)|(2<<14)|)$

Set "10"

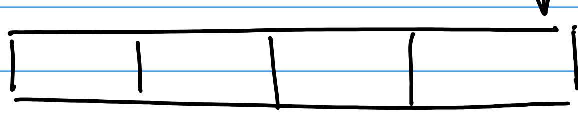
Note: CR0, CR1

LPC_SSP1 → CR0

Control Register

Table 371

CR0



Tech Spec.

1° 8 bit Transfer.

CR0[3:0] = 0111 = 0x7

2° SPI

CR0[5:4] = 00 (SPI)

3° Clock f_{SPI}

a. $\underbrace{\text{CR0}[15:8]}_{8 \text{ bits}}$ $255 = 2^8$

b. $f_{SPI} = \frac{PCLK}{(SCR+1)\, CPSDVSR}$ ...(1)

[0, 255]

mauch2and.

Today's Topics:

1° Midterm Review

2° SPRs CR0, CR1 for SPI I/F

Ref:

1° CPU Datasheet. pp 431-433.

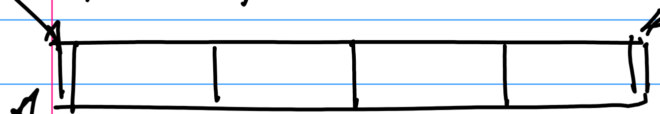
2° Sample code SPI init (Draw a Line)

Example: SSP.C Source Code Walk-Through. 151-208

line 162-165 PINSEL0

Table 80, PP. 117 CPU Datasheet

r31 r0



PINSEL0[31:30] = 10

Table 81. PINSEL1

= 0x2A

10 from CPU Datasheet

PINSEL1[1:0], PINSEL1[3:2], PINSEL1[5:4] → SSP1

SSEL0, MOSI0, MISO0

line 173

CR = 0x0707 → Tech Spec.

.... | 0000 | 0111 | 0000 | 0111

16 upper Bits all zeros.

From Datasheet PP431

8 bit Transfer

CR0[15:8] SCR

CR[5:4] = 00 SPI

Clock

$$f_{SPI} = \frac{PCLK}{(SCR+1) * DVR}$$

CR0

CR0[15:8] = 0x7

CR1[2] = 0 for "Master"

$$f_{SPI} = \frac{1 \times 10^6}{(7+1) * DVR}$$

To find f_{SPI}.

$2^8 = 256$, SCR ∈ [0, 255]

PP. 433. CPSDVSR ∈ [2, 254]

Midterm Review

1° Video ON, Mandatory.

a. Submission to CANVAS 15 min. File Uploading No Late Submission After the Deadline paper will be disqualified

b. If CANVAS Disrupted, then E-mail Submission

c. file in "zip"

FirstName + 4 Digits + CMPE242 mid.zip
SID

2° 3 Questions ±

Hardware: CPU Block Diagram, memory map, SFRs.

CKT, SCH Design

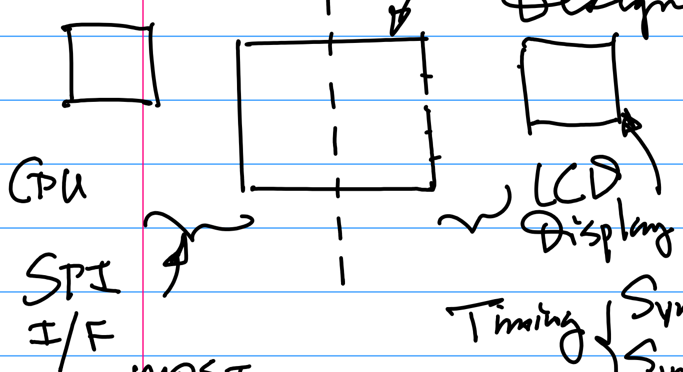
Software: Coding

CMPE240

SPR Coding → Binary Pattern
- for Init & Config
Debugging Purpose ← Tech Spec.

Algorithm:
2D Vector Graphics.
G.E.

a LPC CPU LPC Driver.
LPC Deriver Design



Timing { Sync F, Sync H, Sync D

MOSI
MISO
SCK
SSELn

$\overline{CS}$

b Virtual v.s. physical Display Transform.

c $\vec{P} = \vec{P_i} + \lambda(\vec{P_{i+1}} - \vec{P_i})$

Screen Saver. { i. Rotation! No W/O Rotation Matrix
ii 2D Transforms

Composition of 2D Transform. { R_{3x3}, T_{3x3} → Tree.

Preprocess + R_{3x3} + Post ~

Formula: One Page Formula Sheet; is allowed, However, No Example, or Verbal Explanation is Not Allowed, Submission of the formula Page is required with your mid-term paper.

No multiple choice question.

SCH: Requires All the pins needed in the design to have Label; wire: "Arrow" to indicate direction.

Block Diagram: wire(s), Label(s) direction (Arrow)

CPU Datasheet will be Provided

C Code program will be provided for Answering questions, or for Redesign.

Calculator is allowed;

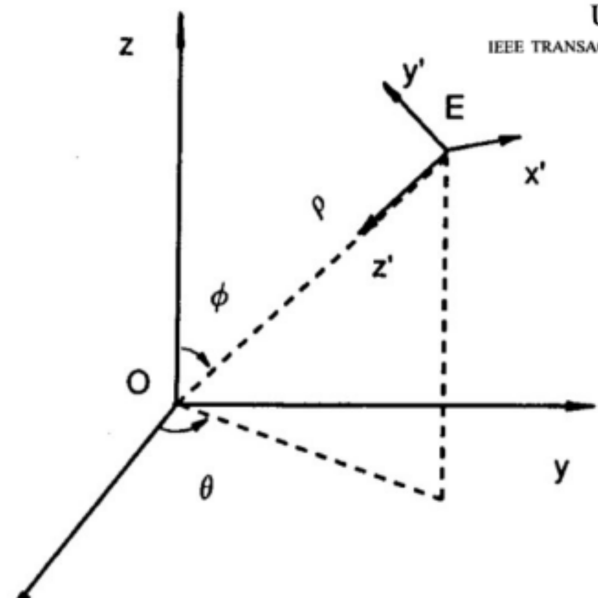
CMPE240

April 5 (Monday)

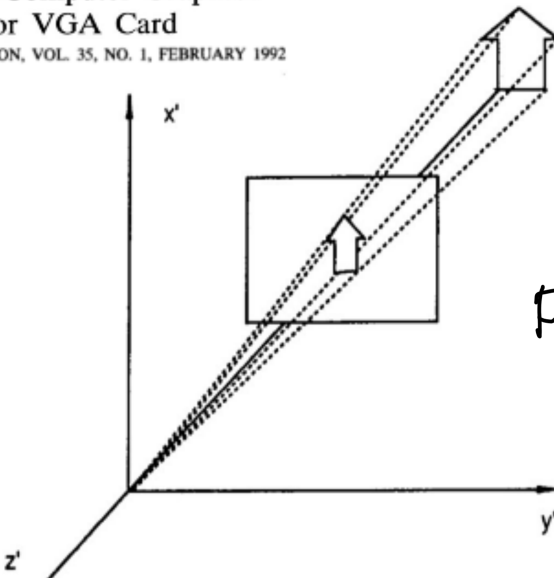
1° Midterm Key on github, "Key" To search.

3D Transformation Pipeline Technique

Reference: H. Li Three-Dimensional Computer Graphics Using EGA or VGA Card
IEEE TRANSACTIONS ON EDUCATION, VOL. 35, NO. 1, FEBRUARY 1992



Step 1. World-to-viewer transform



Step 2. Perspective Projection

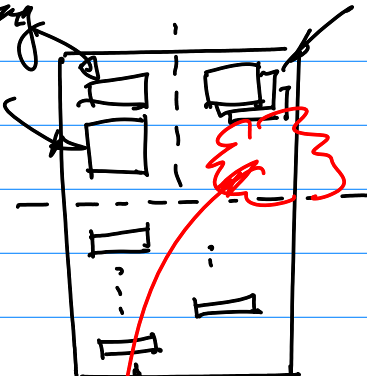
Fig 1.

$$\mathbf{T} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi & 0 \\ -\sin\phi\cos\theta & -\sin\phi\cos\theta & -\cos\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(1)$$

$$x_p = x_e\left(\frac{D}{z_e}\right) \qquad y_p = y_e\left(\frac{D}{z_e}\right) \quad \ldots(2)$$

Harry Li, Ph.D.

Today's Topics: 3D G.E.



Peripheral Controllers

2D G.E.
3D G.E. → GPU

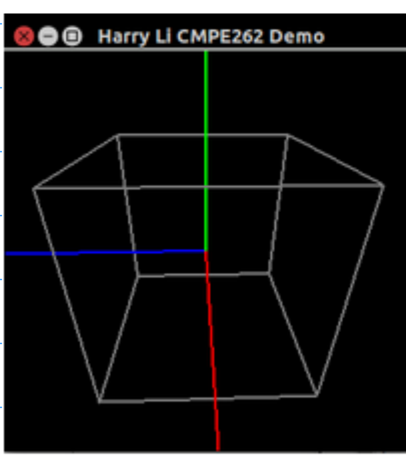


Fig 2a

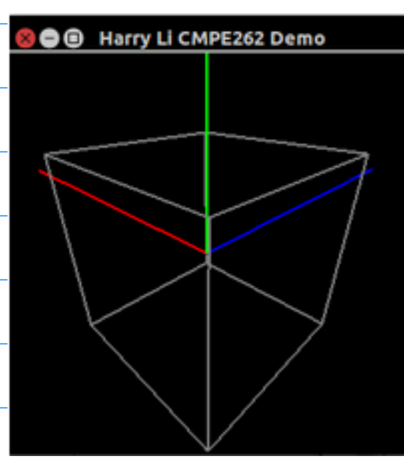


Fig 2b

from github. Key "111a, 111b, ..."

Note: 2D G.E. { Vector Graphics, Transformation, Primitive Graphics { a D.D.A., b line, c Arc/circle etc. } Transistor / Gate Level. }

Example: 3D Wireframe Model

First, X_w-Y_w-Z_w World Coordinate System.

a Right System, r-g-b for X_w, Y_w, Z_w axis

b Transformation Pipeline

{ 1st World-2-Viewer

$$T_{4\times4}: (x_w, y_w, z_w) \to (x_v, y_v, z_v) \quad \cdots (3)$$

2nd Perspective Projection

$$P: (x_v, y_v, z_v) \to (x_p, y_p) \quad \cdots (3b)$$ }

Second. Design of Dataset, e.g. 4 Vertices for X_w-Y_w-Z_w Axis

$\vec{P}(x, y, z)$ 3D pt. in X_w-Y_w-Z_w

Step 1. $\vec{P}(x, y, z) \in \mathbb{R}^3$

for the Origin $\vec{P}_0(x_0, y_0, z_0) = (0, 0, 0) \cdots (4)$

$$\vec{P}_x(x_x, y_x, z_x) = (100, 0, 0), \quad \cdots (4\text{-}1)$$

$$\vec{P}_y(x_y, y_y, z_y) = (0, 100, 0), \text{ and } \cdots (4\text{-}2)$$

$$\vec{P}_z(x_z, y_z, z_z) = (0, 0, 100) \quad \cdots (4\text{-}3)$$

#define X_w_axis ~

⋮

#define Z_w_axis ~

Now, Implementation (Drawing r-g-b axis)

Homework: Draw r-g-b axis on your LPC Display. Bring your Program / Board to the Next Class.

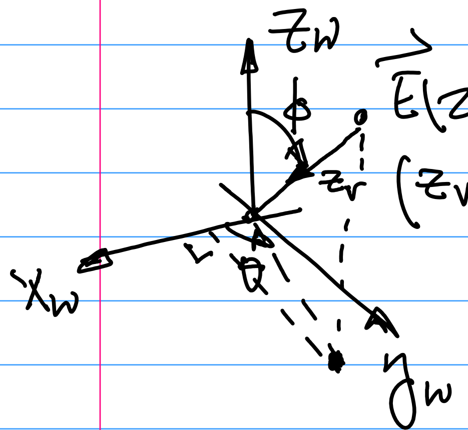
Step 2. World_2_Viewer

$$T_{4\times4}: (x_w, y_w, z_w) \in \mathbb{R}^3 \to (x_v, y_v, z_v) \in \mathbb{R}^3$$

$$\begin{pmatrix} x_v \\ y_v \\ z_v \\ 1 \end{pmatrix} = T_{4\times4} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad \cdots (5)$$

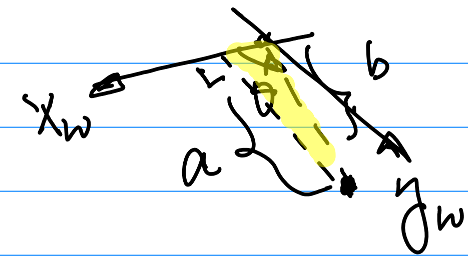
"After" "Before"

Now, find the X_w-axis in Viewer Coordinate



$\vec{E}(200, 200, 200) = (x_e, y_e, z_e)$
$(z_v \Rightarrow z_e)$

Fig 1.



$\sin\theta = \frac{a}{b}$

$a = y_e = 200$

$b = \sqrt{x_e^2 + y_e^2} = 200\sqrt{2}$

$\therefore \sin\theta = \frac{200}{200\sqrt{2}} = \sqrt{2}/2$

$\cos\theta = \frac{x_e}{b} = \frac{200}{200\sqrt{2}} = \sqrt{2}/2$

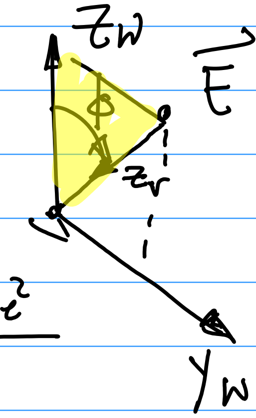
for ϕ (phi)

$\cos\phi = \frac{z_e}{\rho}$

$\rho = \sqrt{x_e^2 + y_e^2 + z_e^2}$

"rho"

$\sin\phi = \frac{\sqrt{x_e^2 + y_e^2}}{\rho}$



a. Vector Passing Through $\vec{E}$, Perpendicular to X_w-Y_w plane, So form an intersection pt.

b. Vector Passing through the intersection pt on X_w-Y_w, perpendicular to X_w axis.

Note: Angle θ (Theta) on X_w-Y_w plane
{ wrt Positive X_w-axis, And
{ Counter Clockwise Direction

Note: Angle ϕ (phi) on z_v-z_w plane.

Now, find each entry on $T_{4\times4}$ matrix,
So World-2-Viewer Transform
Can be Performed.

for Angle θ (Theta)
$\sin\theta$, and $\cos\theta$