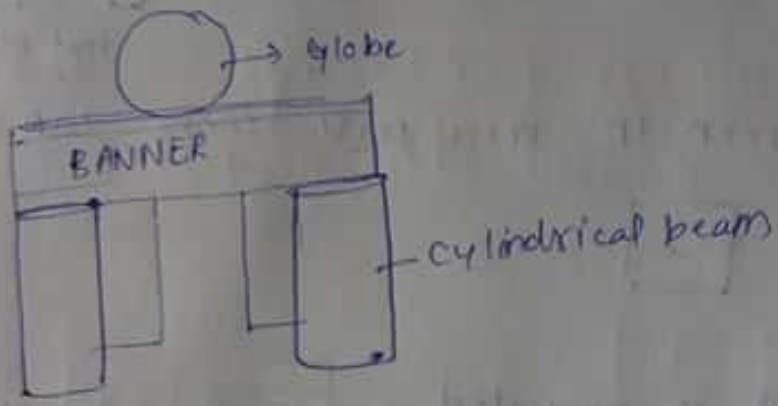
1. Basic Primitives

Sphere/Ellipse/Circle → for the globe

Rectangle → pillars and all other items

Cylinder → cylindrical beam

2. Front view



Top view

3. Graphics pipeline.

The steps are

MODEL → WORLD → CAMERA → VIEWPORT → SCREEN

1) MODEL:

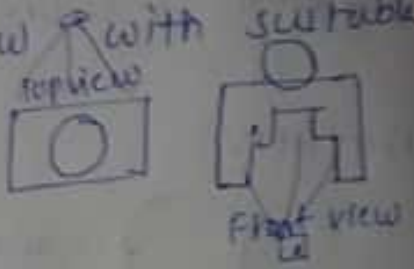
To create the above model we require sphere (globe), two cylinders for 2 cylindrical beams and Rectangular surfaces for LED display.

WORLD:

The model is placed in the world. The model then gets the world coordinates. Here we can adjust size, scaling rotation various operations can be performed

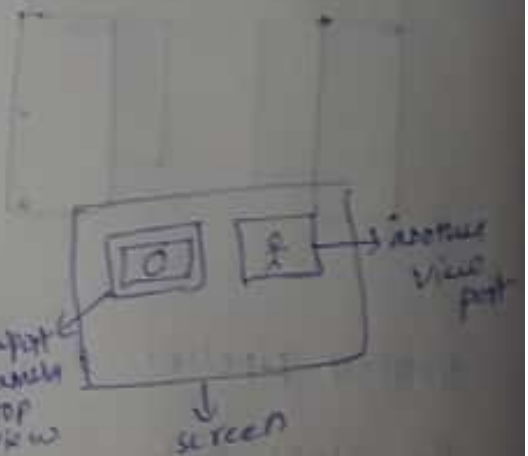
Camera:

It is for getting a particular view with suitable position like top view, front view.



VIEW PORT

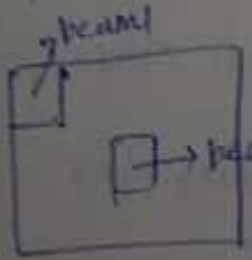
These are shown in screen In view ports there can be multiple view ports.



Screen:

It is where the viewport is projected.

To see beam 2 at (300, 300) we need to translate



by translation

$P' = P + T$

P' - new coordinate

P - old coordinate

T - translation vector

In homogeneous coordinate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

x', y' are new coordinates

x, y are old coordinates

$(tx, ty) = (300, 300)$

$$P = \begin{bmatrix} tx \\ ty \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 100 \\ 300 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 300 \\ 200 \end{bmatrix}$$

④ The canvas should be in ratio 1:1 to view that clearly.

a) Aspect ratio

b) camera position

c) Depth of field.

d) Lightening

e) exposure

f) field of view

c) Rendering pipeline.

Rendering

projection - occulusion(lighting exposure) : colour

shadows

After rendering process a vector image is procured which is comprised of points & paths rather than pixels thus

containing the image of building in top or front view as defined with all our requirements

g) This can be done by ray tracing which is a rendering technique for generating an image by tracing the path of light as pixels in an image plane and simulating the effects of its interaction with vertical objects

h) by pushMatrix() and popMatrix(), we can apply transformations on beam 2. without effecting beam 1. pushMatrix saves the current coordinate system in stack where as popMatrix restores it

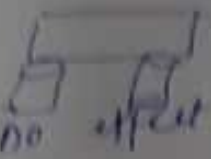
i. $\begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{bmatrix} = M_{modul} \begin{bmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ 1 \end{bmatrix}$

$\begin{bmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ 1 \end{bmatrix} = M_{modulview} \begin{bmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{bmatrix} = M_{view} M_{modul} \begin{bmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{bmatrix}$

$\begin{bmatrix} x_{clip} \\ y_{clip} \\ z_{clip} \\ w_{clip} \end{bmatrix} = M_{projection} \begin{bmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{bmatrix}$

k) No, it can't be done unless we restore the picture. As any operation such as adding texture, color, etc can be done only through pixels which is a primitive or raster picture. This can be only found in photoshop where you would rasterize picture for applying some color correct and other operations.

l) Since clipping is cutting a object where you want the clipped part other is excluded



i) when we apply culling to beam 1, it has no effect on beam 2 but beam 1 will be excluded out of pipeline

m) RGB stands for RED, GREEN and BLUE and ranges from 0 to 255 for R, G and B.

i) (0,0,0) → Black colour is filled to beams.

ii) (255,255,255) → white color is filled to beams.

So we will get black and white for values of R,G,B as mentioned above.