**ASSIGNMENT-1**

**1. COLORS**

**1) Why do we use three values (not four or two) to represent a color in computer systems?**

**A)** Everything in a computer is represented by numbers. In fact, everything is represented by zero and one. Binary numbers use zeros and ones to represent numbers. In computers, electronic switches are used to represent numbers. These switches are also referred to as "bits." These bits are stored in the memories of our computers. A "byte" is a unit of data made up of eight bits. Colors are most commonly represented in computers using 8-bit values. This means that a specific color component is represented by a set of eight ones and zeros. Eight 0s and eight 1s can be combined in any order to produce 256 different color levels. For example, the decimal integer 0 is represented by the 8-bit binary digits 00000000.while the decimal integer 255 is represented as 11111111.

Colors can be numerically expressed in a variety of ways. The most common method used in computers is to represent the ratio of red, green, and blue primary lights required to produce the desired colors. As a result, black would be represented by (R=0, G=0, B=0) and white by (255,255,255). The values (255,0,0), (0,255,0), and (0,0,255) correspond to the red, green, and blue primaries, respectively. Similarly, the secondary colors would be represented by (0,255,255), (255,0,255), and (yellow, magenta, and cyan) (255,255,0).

The RGB color model has been used in traditional photography, but its primary application is in electronic devices such as televisions and computers for sensing, representing, and displaying images. Inside the human eye, there are millions of light sensors, and "cones" are the ones that recognize color. Cones can distinguish between red, blue, and green light. An object's color is determined by the amount of red, blue, and green light it contains. Computer screen pixels take advantage of this by emitting the amounts of red, blue, and green light that your eyes will interpret as the desired color. As a result, when you see "purple," your eyes' red and blue cones light up are activated, which your brain interprets as a perceived color. Because a color is made up entirely of the primary colors (red, green, and blue), there are only three ways to describe how much of each primary color is present in each color. With 256 possible values for each of the three fundamental colors, there are more colors than the human eye can distinguish. This results in a total of 16,777,216 possible colors (256 x 256 x 256).

**2) In OpenGL, colors can be represented by 4 dimensional tuples in the OpenGL, such as glColor4f(a,b,c,d). Please explain the meaning of a, b, c, and d.**

**A)** glcolor4f(a,b,c,d): The meaning of a,b,c and d’s are

a is the component of red color

b is the component of green color

c is the component of blue color

d is alpha intensity

Internal normalization is performed on each parameter between 0.0 and 1.0. The size 3 float array containing x, y, and z co-orders is anticipated by the two APIs.

glColor4f(1, 0, 0, 0.5);

for example, in OPENGL the triangle can be drawn as

glBegin(GL\_TRIANGLES);

glColor3f( 1, 0, 0 ); // red

glVertex2f( -0.8, -0.8 );

glColor3f( 0, 1, 0 ); // green

glVertex2f( 0.8, -0.8 );

glColor3f( 0, 0, 1 ); // blue

glVertex2f( 0, 0.9 );

glEnd();

**3) What is color system? What are the features of color systems?**

**A)** A color system is a collection of colors that define a particular range of visual perception. These colors are mixed to create a limited range of useful colors. In addition to RGB, CMYK, and LUT are examples of color systems.

The components of color systems are

1. Dominant frequency (or) Hue
2. Saturation (or) Chroma
3. Lightness (or) Value

**1.HUE:** The most distinguishing feature of a color is its hue. There are an infinite number of possible colors. For example, every possible orange hue exists between red and yellow. It is measured in degrees circumnavigating horizontal circles. A dominant frequency of light is perceived at the red end of the spectrum in the case of red objects.

**2.CHROMA:** The purity and intensity of a color are referred to as saturation. Colors that are saturated appear more rich and complete. Desaturated colors might look washed out or uninteresting. While rich jewel tones are extremely saturated, pastel hues are desaturated.

**3.VALUE:** The color's lightness or darkness is what gives it its value. Dark hues are frequently referred to as shadow, and light ones as pastel. Luminance is another name for this quantity.

Chart, sunburst chart

Description automatically generated

**2. IMAGE AND VOLUME**

**(1) If we use 16 bits to represent each channel of a color with three channels, what is the memory size used to store a 1024\*1024 color image?**

**A) Given, 16 Bits = 6 Bytes because 16bits=1 byte**

**Therefore, bits range=2^16-1=65535 (0,65535)**

**Bits per channel = 16 bpc**

**Bits per pixel = 48 bpp**

**Total memory size=1024\*1024\*48**

**=50331648**

**~=6MB**

**(2) If we use 16 bits to represent density in a volume data, what is the memory size of a 1024\*1024\*1024 volume?**

**A) Given, density is measured is 16 bits.**

**Volume=1024\*1024\*1024**

**We can write it as, volume = 2^10\*2^10\*2^10=2^30**

**(since, 1024 can be written as 2^10)**

**Therefore, memory size =2^30-16**

**=2^14**

**=2^10 \*2^4 (--2^10=1k)**

**=16K**

**3. The viewing frustum volume in a graphics pipeline is defined by top, bottom, left, right, near, and far clipping planes.**

**(1) What is the purpose for defining the viewing volume?**

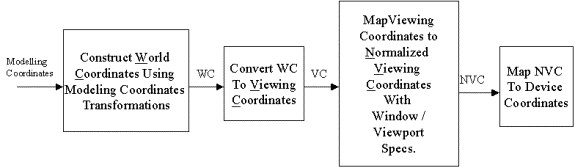
**A)** The mapping of the variables and contours that shape the image into appropriate coordinates at the display device is known as viewing transformation. We are aware that the image is saved in the computer's memory using the World Co-Ordinate System, a practical Cartesian co-ordinate system (WCS). But when an image is displayed on a display device, it is measured using the Physical Device Coordinate System (PDCS) that is specific to that device. As a result, in order to display a picture of a photo, it is necessary to map the coordinates of the points and lines that form the image to the coordinate of the wonderful physiological gadget. This coordinate mapping is carried out using coordinate transformation, also known as viewing transformation.

The viewing transformation which maps photo co-ordinates in the WCS to show co-ordinates in PDCS is performed via the following transformations.

• Converting world co-ordinates to viewing co-ordinates.

• Normalizing viewing co-ordinates.

• Converting normalized viewing co-ordinates to gadget co-ordinates.



The steps involved in viewing transformation: -

* Construct the scene in world co-ordinate the usage of the output primitives and attributes.
* Obtain a particular orientation for the window by means of putting a two-dimensional viewing co-ordinate system in the world co-ordinate airplane and outline a window in the viewing co-ordinate system.
* Use viewing co-ordinates reference body to supply a technique for placing up arbitrary orientations for rectangular windows.
* Once the viewing reference body is established, radically change descriptions in world co-ordinates to viewing co-ordinates.
* Define a view port in normalized co-ordinates and map the viewing co-ordinates description of the scene to normalized co-ordinates.
* Clip all the components of the photograph which lie backyard the viewport.

**(2) What is the difference of the frustum volume between the perspective and orthogonal projection?**

**A) ORTHOGONAL PROJECTION:**

Projections that don't require point of view adjustment are referred to as orthographic or parallel projections. Since there is no adjustment for distance from the camera in these projections, objects on the screen will appear the same size whether they are nearby or far away.

Historically, OpenGL includes this kind of projection for use in CAD, or computer-aided design. Making 2D games or isometric games utilizes orthographic projections in some cases. We utilize the glOrtho() function supplied by OpenGL to put up this type of projection.

glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);

left and right specify the x-coordinate clipping planes, backside and top specify the y-coordinate clipping planes, and close to and some distance specify the distance to the z-coordinate clipping planes. Together these coordinates furnish a container fashioned viewing volume.

Since orthographic projections are frequently used in 2D scenes the Utility Library provides an additional events to set them up for scenes that might not be the use of the z-coordinate.

gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top);

**PERSPECTIVE PROJECTION:**

In perspective projections, an object will get smaller on the screen as it travels further away from the viewer—this effect is known as foreshortening. A frustum, or pyramid with the pinnacle cut off, is the shape of the viewing volume for a standpoint projection, with the narrow end towards the viewer.

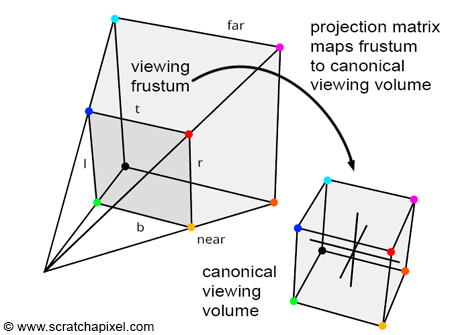
The perspective projection can be set up in a variety of unique ways depending on the view frustum. The first thing we'll look at is this:

void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);

Using glFrustum enables you to specify an asymmetrical frustum, which can be very useful in some instances, but is not what you normally want to do. For a different solution we again flip to the Utility Library:

void gluPerspective(GLdouble fov, GLdouble aspect, GLdouble near, GLdouble far);

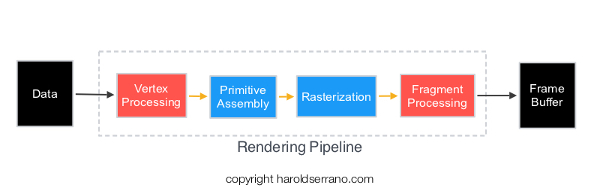
fov specifies, in degrees, the perspective in the y route that is visible to the user; factor is the factor ratio of the scene, which is width divided by way of the height. This will decide the area of view in the x direction.



**4. Graphics pipeline.**

**(1) What is the purpose and task of vertex processing in the pipeline?**

**A)** A pipeline is made up of several data processing devices that are arranged in a chain-like fashion, with the output of one unit being read as the input of the following.

The rendering pipeline, which is another name for the snapshot pipeline, carries out the many stages of the rendering process. Vertex processing, clipping, rasterization, and fragment processing are all included in these regions. The purpose of the snap shots pipeline is to use these four rendering phases to systematize a scene made up of objects, light sources, and a camera and turn it into a two-dimensional image (pixel components). The leftover image presented on the display or display screen is the output of the photo pipeline.

The vertex processor is the initial processing element in the pipeline for images. This processor is in charge of carrying out all geometric operations and computing the shade values for each vertex or component that makes up an object. The vertex processing stage includes three transformations at some point. The geometric specification of three-dimensional world objects serves as the input for the first of these, known as the modelling transformation. Every object that was first specified using local coordinates is subsequently updated to use global coordinates. The global coordinate system has now replaced the unbiased nearby coordinate system for each item. This gives each object access to a common global coordinate space. This transformation stage includes all translations and rotations.

All spatial coordinates are transformed into coordinates given in terms of a viewer's viewing location and orientation in the subsequent transformation step, known as the viewing transformation. A viewer or camera that can be moved and rotated to any location in world coordinate space is the result of this transformation stage. The scene is shown in three dimensions from the viewpoint of this viewer. In view space, both culling and clipping are done.

In order to display the three-dimensional picture on a flat surface, the last transformation, known as the projection transformation, converts the coordinates from view space to two-dimensional image space, or screen space. **(2) Rasterization is an important step. Explain the task and basic techniques used in rasterization?**

**A)** Rasterization is the process of converting a vertex representation to a pixel representation. It is also known as scan-conversion. Each triangle in our model is transformed (rotated, etc.) and projected by the transformation and projection matrices. Next, we cut out each triangle on the image plane. Each triangle is completely inside, completely outside, partially visible with respect to the image plane. We will process the triangle in order of scanning: from left to right starting in the upper left corner, going right and down. We sort the vertices by their y-values ​​and find the vertex with the maximum y-value; Name this node v0. This sorting allows us to identify the other two vertices, v1 and v2. Using the edge slopes, we can calculate each line of pixels to be processed, called a pixel span. On each polygon, we interpolate different di data values ​​for each pixel.

Example: RGB to assign colors to vertices

Chart, line chart

Description automatically generated

**(3) There are four major data components need to be specified before image formation. What are the four components?**

**A)** Mimic is the synthetic camera model for designing graphics in both hardware and software. For this we must specify the four main components before image formation are

• Objects

• Materials

• Viewer

• Light source

Other information (input)we can get from devices such as mouse, keyboard and

system capabilities.

OBJECTS:

Most APIs support a limited set of primitives, including:

* Points (object 0D), Line segments (1D objects), Polygons (2D objects), Some curves and planes, Quadrics, Parametric polynomials.
* They are all defined by positions in space or vertices.

Example:

glBegin(GL\_POLYGON)

glVertex3f(0.0, 0.0, 0.0);

glVertex3f(0.0, 1.0, 0.0);

glVertex3f(0.0, 0.0, 1.0);

glEnd( );

MATERIALS:

It has many properties like

* Absorption
* Scattering

Absorption follows the same properties as color. And Scattering includes the diffuse or specular.

VIEWER:

Viewer should have specifications like position, orientation, focallength and film plane.

Diagram

Description automatically generated

LightSource:

There are different types of lights. Such as

* Point sources vs distributed sources

Distributed sources can be defined as if the lightsource was infinitely far away from the surface that it is illuminating. Sunlight is the best example for distributed sources.

Point sources can be defined as the light originates from a point. The point may be infinity. Approximation for light sources whose dimensions are small relative to objects.

* Spotlights

a projected spot of light used to illuminate brilliantly a person, object, or group on a stage

* Near and far sources
* Color properties