COMPUTER GRAPHICS ASSIGNMENT

B. Tech CSE - Open Source and Open Standards

Submitted to :

Mr. Arjun Arora

Submitted By:

Name : KUNAL BHARDWAJ

Roll no : 027

School of Computer Science

University of Petroleum and Energy Studies,

Dehradun – 248007: Uttarakhand

Q1. Explain the working of CRT? Also what happens in case of colour CRT.

Ans 1 . The working of CRT depends on the movement of electrons beams. The electron guns generate sharply focused electrons which are accelerated at high voltage. This high-velocity electron beam when strikes on the fluorescent screen creates luminous spot, After exiting from the electron gun, the beam passes through the pairs of electrostatic deflection plate. These plates deflected the beams when the voltage applied across it. The one pair of plate moves the beam upward and the second pair of plate moves the beam from one side to another. The horizontal and vertical movement of the electron are independent of each other, and hence the electron beam positioned anywhere on the screen.

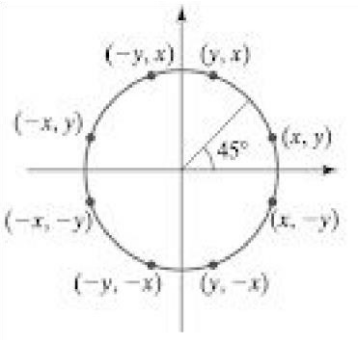
The working parts of a CRT are enclosed in a vacuum glass envelope so that the emitted electron can easily move freely from one end of the tube to the other.

COLOR CRT

In case of color CRT, the electrons are generated by multiple electron guns (typically three producing the primary colours of RED, GREEN and BLUE) located in the narrow end of the CRT. The source of the electrons is an oxide coated element called the cathode. The cathode must be heated to produce the required stream of electrons (negatively charged) that are then attracted to the screen — which is positively charged. The yoke — a coil that is mounted on the tube's neck — is synchronized with the video input signal (from the computer) and is responsible for the synchronized horizontal and vertical movement of the image on the screen.

Q2. Explain Circle Generation Algorithm.

Ans 2.



If Radius is 'r' and circle center is (x_c,y_c) then

First point on circumference of circle centered at origin is : $(x_0,y_0) = (0,r)$

Initial value of decision parameter , $P_0 = 1-r$

Test Condition :

Case 1 :

If $P_k < 0$
Point to plot $(x_k + 1, y_k)$; and $P_{k+1} = P_k + 2x_{k+1} + 1$

Case 2 :

If $P_k >= 0$
Point to plot $(x_k + 1, y_k - 1)$; and $P_{k+1} = P_k + 2x_{k+1} + 1 - 2(y_{k+1})$

Note : Condition Continues till x >=y

Q3. Discuss the rendering pipeline in OpenGL.

Ans3. The sequence of steps that OpenGl takes when rendering objects is termed as a rendering pipeline.

Steps :

1. **Vertex Specification** : Ordered list of vertices that define the boundaries of the primitives. Attributes like color,texture coordinates can also be defined.
2. **Vertex Shader** : It calculates the final position of each vertex. It i s a program written in GLSL that manipulates vertex data.
3. **Tessellation** : This is an optional stage. In this stage primitives are tessellated i.e. divided into smoother mesh of triangles.
4. **Geometry Shader** : Geometry Shader takes an input primitive and generates zero or more output primitive. It can also convert primitives to different types. For example, point primitives can become triangles.

5. **Vertex Post Processing** :This is a fixed function stage i.e. the user has a very limited to no control over these stages. It performs Clipping that discards the area of primitives that lie outside the viewing volume
6. **Primitive Assembly** : This stage collects the vertex data into an ordered sequence of simple primitives(lines, points or triangles).
7. **Rasterization** : it produces the output in fragments.
8. **Fragment Shader** : The job of the fragment shader is to determine the final color for each fragment. It runs for each fragment in the geometry.
9. **Per-Sample Operations** : There are few tests that are performed based on user has activated them or not. Some of these tests for example are Pixel ownership test, Scissor Test, Stencil Test, Depth Test.

Q4. Draw a line using Bresenham Algorithm having endpoints (20,10) and (30,18).

Ans 3.

$\Delta x = x2-x1 = 10$

$\Delta y = y2 - y1 = 8$

$2\Delta y = 16$

$2\Delta x = 20$

$2\Delta y - 2\Delta x = -4$

At starting pixel position is (x_0 , y_0) with decision parameter P_0 given by $P_0 = 2\Delta y - \Delta x$ hence ,

$P_0 = 6$

I. $P_0 > 0$, Point to plot (21,11)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 6 + 16 - 20 = 2$

II. $P_1 = 2$, Point to plot (22,12)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 2 + 16 - 20 = (-2)$

III. $P_2 = -2$, Point to plot (23,12)

$P_{k+1} = P_k + 2\Delta y = -2 + 16 = 14$

IV. $P_3 = 14$, Point to plot (24,13)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 14 + 16 - 20 = 10$

V. $P_4 = 10$, Point to plot (25,14)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 10 + 16 - 20 = 6$

VI. $P_5 = 6$, Point to plot (26,15)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 6 + 16 - 20 = 2$

VII. $P_6 = 2$, Point to plot (27,16)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 2 + 16 - 20 = (-2)$

VIII. $P_7 = (-2)$, Point to plot (28,16)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = -2 + 16 = 14$

IX. $P_8 = 14$, Point to plot (29,17)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 14 + 16 - 20 = 10$

X. $P_9 = 10$, Point to plot (30,18)

$P_{k+1} = P_k + 2\Delta y - 2\Delta x = 10 + 16 - 20 = 6$

k	P	Point X	Point Y
0	6	21	11
1	2	22	12
2	-2	23	12
3	14	24	13
4	10	25	14
5	6	26	15
6	2	27	16
7	-2	28	16
8	14	29	17
9	10	30	18