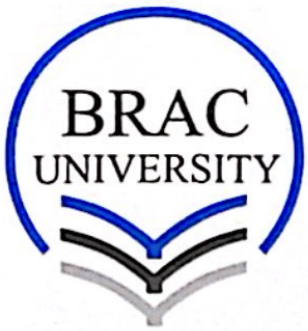

Inspiring Excellence

CSE423: Computer Graphics

Assignment 01

Submitted by,

Mohammad Jabir Safa Khandoker

ID: 22201108

Section: 13

Department: CSE

Submitted to,

Mehnaz Ara Fazal (MZFZ)

Computer Science and Engineering (CSE) Department

BRAC University

Date of Submission: 02 March 2025

Ans to the Ques. No. 01

(a)

$(5, -15)$ to $(0, -5)$

$d_x = 0 - 5 = -5 < 0$

$d_y = -5 - (-15) = 10 > 0$

$|d_y| > |d_x|$

$\therefore$ Zone 2 : $(-y, x)$

Converting to Zone 0 : (x, y)

Start : $(5, -15) \longrightarrow (-15, -5)$

End : $(0, -5) \longrightarrow (-5, 0)$

$\therefore d_x' = -5 - (-15) = 10 > 0$

$\therefore d_y' = 0 - (-5) = 5 > 0$

$\therefore d_{init} = 2d_y - d_x = 0$

$\therefore \Delta d_{NE} = 2d_y - 2d_x = -10$

$\therefore \Delta d_E = 2d_y = 10$

Zone 0			ΔS		Original zone
x	y	d	d updating	E/NE	Pixel $(-y, x)$
-15	-5	0	d+10 = 10	E	(5, -15)
-14	-5	10	10-10 = 0	NE	(5, -14)
-13	-4	0	0+10 = 10	E	(4, -13)
-12	-4	10	10-10 = 0	NE	(4, -12)
-11	-3	0	0+10 = 10	E	(3, -11)
-10	-3	10	10-10 = 0	NE	(3, -10)
-9	-2	0	0+10 = 10	E	(2, -9)
-8	-2	10	10-10 = 0	NE	(2, -8)
-7	-1	0	0+10 = 10	E	(1, -7)
-6	-1	10	10-10 = 0	NE	(1, -6)
-5	0	0	0+10 = 10	E	(0, -5)

(b)

$(5, -15)$ to $(0, -5)$

$$\therefore m = \frac{-5-(-15)}{0-5} = \frac{10}{-5} = -2 \quad B < -1$$

y	x	Rounded x	Pixel
-15	5	5	(5, -15)
-14	4.5	5	(5, -14)
-13	4	4	(4, -13)
-12	3.5	4	(4, -12)
-11	3	3	(3, -11)
-10	2.5	3	(3, -10)
-9	2	2	(2, -9)
-8	1.5	2	(2, -8)
-7	1	1	(1, -7)
-6	0.5	1	(1, -6)
-5	0	0	(0, -5)

Ans to the Ques. No. 02

(a)

Given,

Resolution = 3840 × 2160

∴ Total Pixels = Width × Height = 3840 × 2160

= 8,294,400 pixels in one frame.

(b)

Given,

Frame Rate = 60 fps

∴ Frame Time = $\frac{1}{\text{Frame Rate}} = \frac{1}{60}$

= 0.01667s = 16.67 ms

(c)

Given,

For GPU, N = 50,000 pixels

Now, GPU Frame Time = $\frac{\text{Pixels Per Frame}}{N} = \frac{8294400}{50000}$

≈ 165.888 ms > Frame Time 16.67 ms

∴ GPU can't render the entire frame.

Ans. to the Ques. No. 03

$x_{max} = 60$ $\quad$ $y_{max} = 100$

$x_{min} = -50$ $\quad$ $y_{min} = -10$

$x_1 = -50$; $x_{min} \leq x_1 \leq x_{max}$ $\quad$ bit 0 = 0, $\quad$ bit 1 = 0

$y_1 = -70$; $y_1 < y_{min}$ $\quad$ bit 2 = 1, $\quad$ bit 3 = 0

$\therefore$ oc1 = 0100

$x_2 = 40$; $x_{min} \leq x_2 \leq x_{max}$ $\quad$ bit 0 = 0, $\quad$ bit 1 = 0

$y_2 = 100$; $y_{min} \leq y_2 \leq y_{max}$ $\quad$ bit 2 = 0, $\quad$ bit 3 = 0

$\therefore$ oc2 = 0000

Now, oc1 AND oc2 =

$$\begin{array}{r} 0100 \\ 0000 \\ \hline 0000 \end{array}$$

$\therefore$ The line is partially inside.

oc1 = 0100 != 0000

The outcode means it is below the clipping window.

∴ For Bottom intersection,

$y = y_{min} = -10$

$x = x_1 + \frac{1}{m}(y_{min} - y_1)$

$= -50 + \frac{90}{170}(-10 - (-70))$

$= -18.235$

$m = \frac{y_2 - y_1}{x_2 - x_1}$

$= \frac{100 - (-70)}{40 - (-50)}$

$= \frac{170}{90}$

∴ $x = -18.235$; $x_{min} \leq x \leq x_{max}$ bit0 = 0, bit1 = 0

∴ $y = -10$; $y_{min} \leq y \leq y_{max}$ bit2 = 0, bit3 = 0

∴ oc1 = 0000

Now, oc1 = oc2 = 0000. ∴ Completely inside.

∴ Line segment (-18.235, -10) to (40, 100) is now completely inside the clipping window.

In Cohen-Sutherland Algorithm, the AND operation checks if the line segment ever passes through the clipping window.

If a bitwise AND operation returns 1 then both bits were valued 1. So, the line segment is ~~either~~ completely out of bounds of the clip window by being either completely above / below / right / left (based on bit index 3, 2, 0, 1) of the clipping window.

Else, the ~~Bo~~ AND operation returns 0 (one bit is 0 and other is 1) meaning the line segment passes through the clip window and is partially inside. So clipping is required.

The AND operation is very simple to execute in the ALU, thus simplifying calculations and optimizing the whole algorithm.