

UNIVERSITI SAINS MALAYSIA

First Semester Examination
Academic Session 2015/2016

January 2016

MSG 387 – Computer Graphics
[Grafik Komputer]

Duration : 3 hours
[Masa : 3 jam]

Please check that this examination paper consists of TWELVE pages of printed materials before you begin the examination.

[Sila pastikan bahawa kertas peperiksaan ini mengandungi DUA BELAS muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]

Instructions: Answer **FIVE** (5) questions.

Arahan: Jawab **LIMA** (5) soalan.]

The question papers **shall not be taken out** from the examination hall and will be collected by invigilators.

Kertas soalan ini tidak boleh dibawa keluar daripada dewan peperiksaan dan akan dikutip oleh pengawas peperiksaan.

In the event of any discrepancies, the English version shall be used.

[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi Bahasa Inggeris hendaklah diguna pakai].

1. Given a true-colour RGB raster screen with resolution of 800×600 (width \times height), the width of the display area measures 12 inches and the aspect ratio is 1. The refresh rate is 60 frames per second (fps). Answer the following questions,
- What is the pixel depth of the screen?
 - How many distinct colour choices are available on the system?
 - How many distinct colours can be displayed at the same time on the screen?
 - How much storage (in megabyte, MB) do we need for the frame buffer?
 - What is the transfer rate (in MB per second) of the datum in the frame buffer to the raster display?
 - How much time (in seconds) is taken to load one frame?
 - How much time (in microseconds) is spent scanning across each row of pixels during screen refresh?
 - How many pixels can be accessed per second?
 - What is the access time (in nanoseconds) per pixel?
 - What is the diameter of a pixel in inches? Assume the pixel is round in shape.

[100 marks]

1. *Diberi suatu RGB skrin raster yang berwarna-sejati dengan leraian 800×600 (lebar \times tinggi), lebar kawasan paparan diukur sebagai 12 inci dan nisbah aspek adalah 1. Kadar segar-semula ialah 60 bingkai per saat (fps). Jawab soalan-soalan di bawah,*
- Apakah kedalaman piksel skrin tersebut?*
 - Berapakah pilihan warna yang berbeza yang terdapat dalam sistem tersebut?*
 - Berapakah pilihan warna yang berbeza yang dapat dipaparkan ke atas skrin tersebut pada masa yang sama?*
 - Berapa besarkah simpanan (dalam megabait, MB) yang diperlukan oleh penimbal bingkai tersebut?*
 - Apakah kadar pemindahan (dalam MB per saat) bagi data-data dalam penimbal bingkai ke paparan raster?*
 - Berapa lama (dalam saat) yang diambil untuk memuatkan satu bingkai?*

...3/-

- (g) *Berapa lama (dalam mikrosaat) yang diambil untuk mengimbas setiap baris piksel semasa penyegaran paparan?*
- (h) *Berapa piksel yang boleh diakses per saat?*
- (i) *Apakah masa akses (dalam nanosaat) bagi satu piksel?*
- (j) *Apakah diameter untuk satu piksel dalam ukuran inci? Anggapkan piksel tersebut adalah berbentuk bulat.*

[100 markah]

2. Mark True (T) or False (F).

- (a) Physical input devices refer to devices which receive input from mice and keyboards only. Thus, voice input device is not considered as a type of physical input devices.
- (b) HSV colours are additive because their primary colours are added to form new colours.
- (c) OpenGL is an event driven programming language, that means when an event is detected, the respective callback function will be triggered.
- (d) The original Bresenham Algorithm for circle is more efficient than Midpoint Circle Drawing Algorithm in circle generation, because the original Bresenham algorithm works well on implicit equations, i.e. $F(x,y)=0$.
- (e) In geometric transformation, the result of rotation then translation is different from translation then rotation.
- (f) In scan-line polygon fill algorithm, topological or edge-shortened method is used when the scan line passes through the polygon vertices.
- (g) In line clipping, slope intercept method is less efficient than Liang-Barsky algorithm because slope intercept method uses region codes and performs unnecessary clipping.
- (h) In two dimensional viewing, zoom in and zoom out effects can be achieved by changing the size of the clipping window.
- (i) `glLoadIdentity()` is the function to assign identity matrix to OpenGL's matrix stacks.
- (j) OpenGL routines `glClearColor()` is used to set the color of the objects we want to draw.

[100 marks]

...4/-

2. *Tanda Betul (T) atau Salah (F).*

- (a) *Peranti-peranti input fizikal merujuk kepada peranti-peranti yang menerima input dari tetikus dan kekunci sahaja. Maka, peranti input suara tidak dianggap sebagai suatu jenis peranti input fizikal.*
- (b) *Warna HSV adalah bersifat penambahan kerana warna-warna prima ditambah untuk mendapat warna-warna yang baru.*
- (c) *OpenGL adalah bahasa pengaturcaraan berpandu peristiwa, iaitu suatu fungsi panggil-balik yang tertentu akan dipanggil apabila peristiwa yang berkaitan dikesan.*
- (d) *Algoritma Bresenham asal untuk bulatan adalah lebih cekap daripada Algoritma Melukis Bulatan Titik-tengah dalam penjaan bulatan kerana Algoritma Bresenham Asal bertindak baik dengan persamaan tersirat, i.e. $F(x,y)=0$.*
- (e) *Dalam transformasi geometri, hasil pemutaran kemudian translasi adalah berbeza dengan translasi kemudian pemutaran.*
- (f) *Dalam algoritma garis-imbias penisian poligon, kaedah topologi atau pemendekan-tepi digunakan apabila garis-imbias tersebut melangkahi verteks-verteks poligon.*
- (g) *Dalam pemotongan garis, kaedah pintasan cerun kurang berkesan daripada algoritma Liang-Barsky kerana kaedah pintasan cerun menggunakan kod-kod kawasan dan melaksanakan pemotongan yang tidak diperlukan.*
- (h) *Dalam pemandangan dua matra, kesan-kesan mengezum masuk dan keluar boleh dicapai dengan menukar saiz tettingkap pemotongan.*
- (i) *`glLoadIdentity()` adalah fungsi untuk mengumpuk matriks identiti kepada matriks-matriks OpenGL.*
- (j) *Rutin OpenGL `glClearColor()` digunakan untuk menetapkan warna objek yang dilukis.*

[100 markah]

3. Refer to Figure 1.

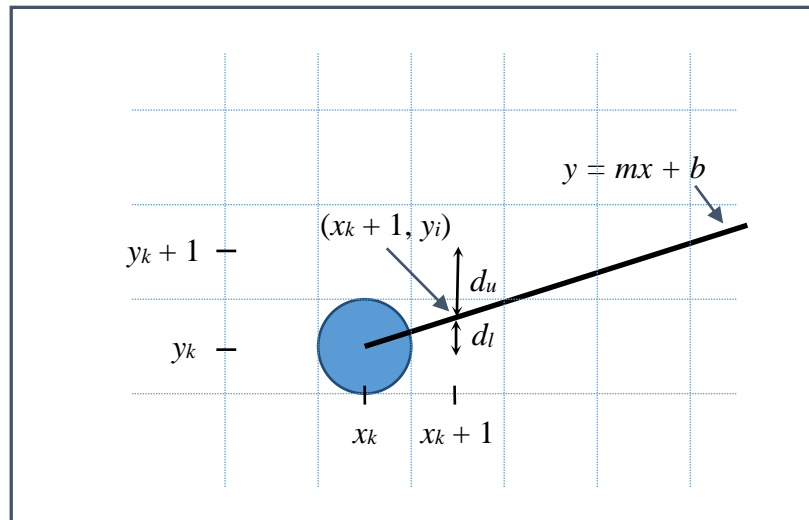


Figure 1.

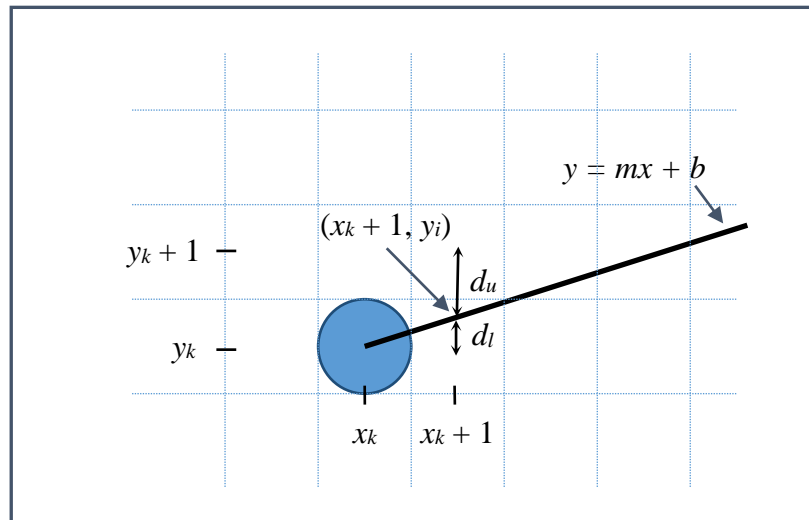
$y = mx + b$ is the straight line equation with one end point at (x_k, y_k) . $(x_k + 1, y_i)$ is a point on the straight line. Figure 1 shows an enlarged region of a raster screen, with the solid circle as a pixel. The pixel's coordinate is given as (x_k, y_k) . d_u is the distance between $(x_k + 1, y_i)$ and $(x_k + 1, y_k + 1)$, d_l is the distance between $(x_k + 1, y_i)$ and $(x_k + 1, y_k)$.

Given x_k, y_k, x_k+1, y_k+1 are known values, we want to find if the next pixel of (x_k, y_k) is $(x_k + 1, y_k)$ or $(x_k + 1, y_k + 1)$.

- Describe Bresenham Algorithm for drawing a straight-line, for the case of $0 < m < 1$, where m is the slope of the straight-line, to find the next pixel of (x_k, y_k) .
- Bresenham Algorithm can be used indirectly to draw a straight line for the case of $m < -1$. Describe the steps in generating the straight line segment for the case of $m < -1$ by using the end points $(4, 9)$ and $(7, 0)$.
- Find all the intermediate pixels between $(4, 9)$ and $(7, 0)$, by using your answer in (b).

[100 marks]

3. Merujuk kepada Rajah 1.



Rajah 1.

$y = mx + b$ adalah persamaan garis-lurus dengan satu titik hujung berada di (x_k, y_k) . $(x_k + 1, y_i)$ adalah satu titik di atas garis-lurus tersebut. Rajah 1 menunjukkan pembesaran suatu kawasan bagi satu skrin raster, dengan bulatan padu sebagai satu piksel. Koordinat piksel tersebut diberi sebagai (x_k, y_k) . d_u adalah jarak di antara $(x_k + 1, y_i)$ dan $(x_k + 1, y_k + 1)$, manakala d_l adalah jarak di antara $(x_k + 1, y_i)$ dan $(x_k + 1, y_k)$.

Diberi $x_k, y_k, x_k + 1, y_k + 1$ adalah nilai-nilai yang telah diketahui, kita hendak mencari bahawa piksel yang berikutan untuk (x_k, y_k) adalah $(x_k + 1, y_k)$ atau $(x_k + 1, y_k + 1)$.

- Huraikan Algoritma Bresenham untuk melukis suatu garis-lurus, untuk kes $0 < m < 1$, di mana m adalah kecerunan garis-lurus tersebut, untuk mencari piksel yang berikutan bagi (x_k, y_k) .
- Algoritma Bresenham boleh digunakan secara tidak langsung untuk melukis garis-lurus untuk kes $m < -1$. Huraikan langkah-langkah yang boleh diambil untuk menjanakan segmen garis-lurus untuk kes $m < -1$ dengan menggunakan titik-titik hujung $(4, 9)$ dan $(7, 0)$.
- Cari semua piksel di antara $(4, 0)$ dan $(7, 0)$, dengan menggunakan jawapan anda dari (b)

[100 markah]

4. (a) Write the following 2D transformation matrices in 3×3 matrix representations.
- (i) Translation matrix, $\mathbf{T}(t_x, t_y)$.
 - (ii) Rotation matrix about the origin, $\mathbf{R}(\theta)$.
 - (iii) Scaling matrix referring to the origin, $\mathbf{S}(s_x, s_y)$.
- (b) An object is rotated 60° about coordinate $(4, -3)$ in anti-clockwise direction, then scaled by half and double of the original size in the vertical and horizontal directions respectively by referring to coordinate $(3, 4)$, then reflected on y-axis, and finally rotated 90° clockwise about the origin. Compute the composite matrix of the transformations.
- (c) Write an OpenGL program segment by using `glTranslatef`, `glRotatef`, and `glScalef`, to achieve the transformation mentioned in (b).

[100 marks]

4. (a) *Tuliskan matriks-matriks transformasi 2D berikut dalam perwakilan 3×3 matriks.*
- (i) *Matriks translasi, $\mathbf{T}(t_x, t_y)$.*
 - (ii) *Matriks putaran sekitar asalan, $\mathbf{R}(\theta)$.*
 - (iii) *Matriks penskalaan merujuk kepada asalan, $\mathbf{S}(s_x, s_y)$.*
- (b) *Suatu objek diputarakan 60° sekitar koordinat $(4, -3)$ ke arah songsang jam, kemudian diskalakan kepada separuh dan duakali ganda saiz asalnya ke arah menegak dan mendatar masing-masing dengan merujuk kepada koordinat $(3, 4)$, kemudian dipantulkan di paksi-y, dan akhirnya diputarakan 90° ke arah ikut jam sekitar asalan. Kira matriks gubahan untuk transformasi tersebut.*
- (c) *Tuliskan segmen program OpenGL dengan menggunakan fungsi-fungsi `glTranslatef`, `glRotatef`, dan `glScalef`, untuk mendapat transformasi yang dinyatakan di (b).*

[100 markah]

5. (a) The parametric equation of a straight-line segment is given as

$$\begin{aligned}x &= x_0 + u \Delta x, \\y &= y_0 + u \Delta y, \quad 0 \leq u \leq 1\end{aligned}$$

where

$$\begin{aligned}\Delta x &= x_{end} - x_0, \\ \Delta y &= y_{end} - y_0, \text{ and} \\ (x_0, y_0) \text{ and } (x_{end}, y_{end}) &\text{ are the two end points of the straight-line segment.}\end{aligned}$$

Derive Liang-Barsky Algorithm, if x_{Wmin} , x_{Wmax} , y_{Wmin} and y_{Wmax} are left, right, bottom and top borders of the clipping window, respectively. You must list down necessary steps to check the line segment before performing the algorithm.

- (b) Given a triangle with local coordinates A(0, 0), B(4, 0) and C(2, 2). The triangle is enlarged 100 times in both horizontal and vertical directions from its original size; and its origin is placed at (340, 50) in the world coordinate reference frame (CRF).

The triangle is then transformed to a clipping window with viewing CRF which has the 2D view-up vector given as (0, 1) and view origin, P_0 given as (0, 0). The left (x_{Wmin}), right (x_{Wmax}), bottom (y_{Wmin}) and top (y_{Wmax}) borders of the clipping window are 100, 500, 50 and 350 respectively in the world CRF.

Then, the coordinates of the triangle are normalized in the normalized CRF, which has the minimum and maximum values of -1 and 1 respectively, for both x and y directions. The triangle is then clipped in the normalized window by using Liang-Barsky algorithm.

After the clipping, the normalized window is then mapped to a viewport with size 400×300 (width \times height), which is then placed at (0, 50) of a display window with size 800×600 . Figure 2 shows the viewport and the display window.

Answer the following questions from (i) to (x). All matrices have 3×3 dimension.

- (i) Find the transformation matrix from local CRF to world CRF. Calculate the coordinates of the triangle after the transformation, and sketch the triangle in the world CRF after the transformation, add the clipped window, then estimate number of intersection points of the triangle with the clipped window, and the shape of the triangle after clipping.

- (ii) Given the transformation sequence is R·T, where R denotes rotation matrix and T denotes translation matrix, find the transformation matrix from viewing CRF to world CRFs.

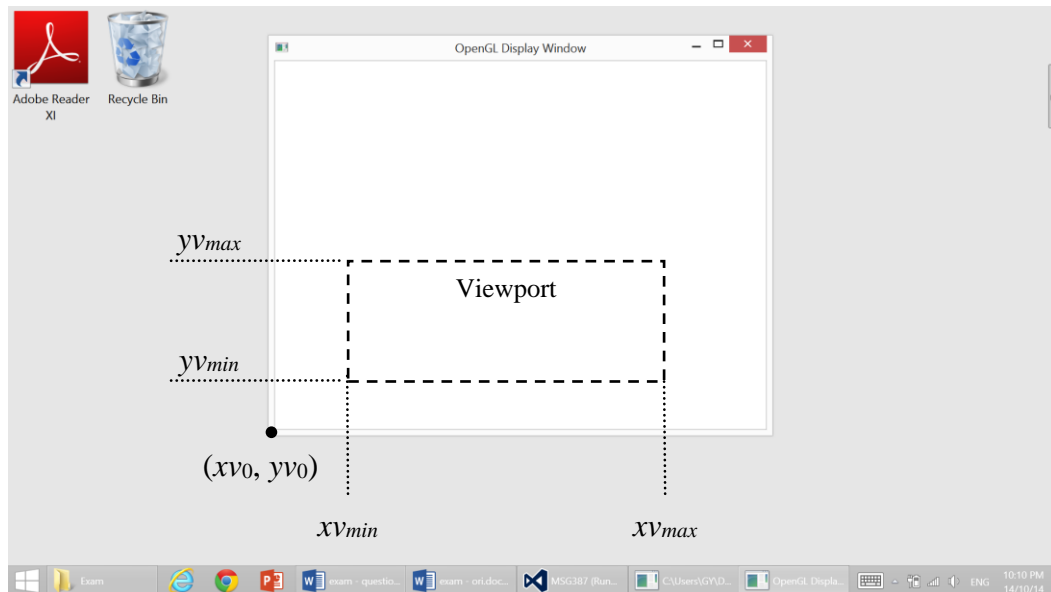


Figure 2. (xv_0, yv_0) is the origin of the viewport; xv_{min} , xv_{max} , yv_{min} and yv_{max} are left, right, bottom and top boundaries of the viewport respectively.

(Note: the drawing is not drawn according to the scale and coordinate).

- (iii) Find the transformation matrix, $M_{window, normsquare}$, from viewing CRF to normalized CRF in terms of xw_{min} , xw_{max} , yw_{min} and yw_{max} . Then compute the value of $M_{window, normsquare}$.
- (iv) Find the coordinates of the triangle in normalized CRF.
- (v) By using Liang-Barsky algorithm, calculate the clipped coordinates of the triangle in normalized CRF.
- (vi) By referring to Figure 2, what are the values of xv_0 , yv_0 , xv_{min} , yv_{min} , xv_{max} , and yv_{max} , measured in viewport CRF?
- (vii) Find the transformation matrix, $M_{normsquare, viewport}$, from normalized CRF to viewport CRF in terms of xv_{min} , xv_{max} , yv_{min} and yv_{max} . Then compute the value of $M_{normsquare, viewport}$.
- (viii) Find the coordinates of the clipped triangle in viewport CRF.
- (ix) Find the device coordinate of the clipped triangle in (viii).
- (x) Sketch the 2D viewing pipeline for the above transformations.

[200 marks]

...10/-

5. (a) *Persamaan berparameter bagi suatu segmen garis-lurus diberi sebagai*

$$\begin{aligned}x &= x_0 + u \Delta x, \\y &= y_0 + u \Delta y, \quad 0 \leq u \leq 1\end{aligned}$$

di mana

$$\begin{aligned}\Delta x &= x_{end} - x_0, \\ \Delta y &= y_{end} - y_0, \text{ dan}\end{aligned}$$

(x_0, y_0) dan (x_{end}, y_{end}) adalah dua titik hujung bagi segmen garis-lurus tersebut.

Terbitkan Algoritma Liang-Barsky, jika xw_{min} , xw_{max} , yw_{min} and yw_{max} adalah sempadan-sempadan kiri, kanan, bawah dan atas untuk tettingkap guntingan masing-masing. Anda mesti senaraikan langkah-langkah yang diperlukan sebelum menjalankan Algoritma Liang-Barsky.

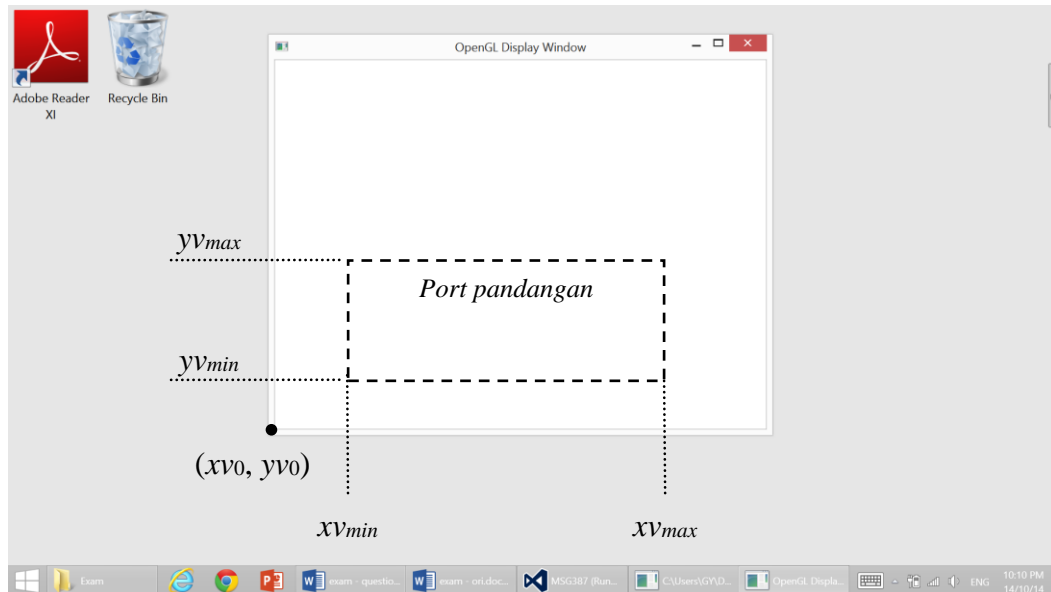
- (b) *Diberi suatu segitiga dengan koordinat-koordinat setempat A(0, 0), B(4, 0) dan C(2, 2). Segitiga tersebut dibesarkan sebanyak 100 kali-ganda ke arah mendatar dan menegak dari saiz asalnya; dan asalannya ditempatkan pada (340,50) dalam kerangka rujukan koordinat (CRF) dunia.*

Segitiga tersebut kemudian ditransformasikan ke suatu tettingkap guntingan dengan CRF pandangan yang mempunyai vektor pandang-tegak 2D diberi sebagai (0, 1) dan asalan pandangan, P_0 diberi sebagai (0, 0). Sempadan-sempadan kiri (xw_{min}), kanan (xw_{max}), bawah (yw_{min}) dan atas (yw_{max}) bagi tettingkap guntingan tersebut adalah 100, 500, 50 dan 350 masing-masing dalam CRF dunia.

Kemudian, koordinat-koordinat segitiga tersebut dinormalkan dalam CRF ternormal, yang mempunyai -1 dan 1 sebagai nilai-nilai minimum dan maksimum masing-masing, untuk kedua-dua arah x dan y . Kemudian, segitiga tersebut digunting dalam tettingkap ternormal dengan menggunakan Algoritma Liang-Barsky.

Selepas guntingan, tettingkap ternormal tersebut dipetakan ke atas suatu port pandangan bersaiz 400×300 (lebar \times tinggi), dan ditempatkan pada (0, 50) bagi suatu tettingkap paparan yang bersaiz 800×600 . Rajah 2 menunjukkan port pandangan dan tettingkap paparan.

Jawab soalan-soalan berikut dari (i) ke (x). Semua matriks adalah berdimensi 3×3 .



Rajah 2. (xv_0, yv_0) adalah asalan bagi port pandangan; xv_{min} , xv_{max} , yv_{min} dan yv_{max} adalah sempadan-sempadan kiri, kanan, bawah dan atas masing-masing bagi port pandangan.

(Perhatian: lukisan di atas tidak dilukis mengikut skala dan koordinat.)

- (i) Cari matriks transformasi dari CRF setempat ke CRF dunia. Hitungkan koordinat-koordinat segitiga selepas transformasi, dan lakarkan segitiga tersebut dalam CRF dunia selepas transformasi tersebut, tambah tettingkap guntingan, kemudian anggarkan bilangan titik persilangan segitiga tersebut dengan tettingkap guntingan, dan bentuk selepas segitiga digunting.
- (ii) Diberi turutan transformasi ialah $R-T$, di mana R ialah matriks putaran dan T ialah matriks translasi. Cari matriks transformasi dari CRF pandangan ke CRF dunia.
- (iii) Cari matriks transformasi, $M_{window, normsquare}$, dari CRF pandangan ke CRF ternormal dalam sebutan xw_{min} , xw_{max} , yw_{min} dan yw_{max} . Kemudian kira nilai bagi $M_{window, normsquare}$.
- (iv) Cari koordinat-koordinat segitiga dalam CRF ternormal.
- (v) Dengan menggunakan Algoritma Liang-Barsky, kira koordinat-koordinat segitiga yang tergantung dalam CRF ternormal.
- (vi) Dengan merujuk kepada Rajah 2, apakah nilai-nilai xv_0 , yv_0 , xv_{min} , yv_{min} , xv_{max} , dan yv_{max} , dalam ukuran CRF port pandangan?
- (vii) Cari matriks transformasi, $M_{normsquare, viewport}$, dari CRF ternormal ke CRF port pandangan dalam sebutan xv_{min} , xv_{max} , yv_{min} dan yv_{max} . Kemudian kira nilai $M_{normsquare, viewport}$.

- (viii) *Cari koordinat-koordinat segitiga yang tergantung dalam CRF port pandangan.*
- (ix) *Cari koordinat peranti bagi segitiga yang tergantung dalam (viii).*
- (x) *Lakarkan talian paip pandangan 2D untuk transformasi-transformatasi di atas.*

[200 markah]