Λαγάρας Στέλιος – ΑΕΜ 1828

Χαρακτηριστικά περιβάλλοντος: Windows 11, WSL2 Ubuntu 20.04, VS Code editor

Για το γραφικό περιβάλλον χρησιμοποιήθηκε το Χ-Lanch (VcXsrv) παραμετροποιώντας την έξοδο display του virtual environment. Βέβαια, αυτό το hack δεν φαίνεται σταθερά λειτουργικό, σε επόμενη έκδοση λέγεται πως θα ενσωματωθεί στο wsl γραφικό περιβάλλον.

Contents

[Άσκηση 1 - Algorithm Liang-Barsky 1](#_Toc96528189)

[Άσκηση 4 – Παράδειγμα Liang-Barsky 1](#_Toc96528190)

[Άσκηση 2 - Cohen Algorithm 2](#_Toc96528191)

[Άσκηση 3 – Παράδειγμα Cohen Algorithm 3](#_Toc96528192)

[Άσκηση 5 - Cohen – Southerland 3D Algorithm 4](#_Toc96528193)

[Άσκηση 6 – Προγραμματιστική 5](#_Toc96528194)

[Άσκηση 7 – Εφαρμoγή του Sutherland-Hodgman 5](#_Toc96528195)

[Exercise 9, RGB – HSV: 9](#_Toc96528196)

## Άσκηση 1 - Algorithm Liang-Barsky

1. Set  and 
2. Calculate the values of tL, tR, tT, and tB [(tvalues)](https://www.cs.helsinki.fi/group/goa/viewing/leikkaus/tvalue.html).
   * if  or  ignore it and go to the next edge
   * otherwise classify the **t**value as [entering or exiting](https://www.cs.helsinki.fi/group/goa/viewing/leikkaus/enter.html) value (using inner product to classify)
   * if **t** is entering value set ; if **t** is exiting value set 
3. If  then **draw a line** from (x1 + dx\*tmin, y1 + dy\*tmin) to (x1 + dx\*tmax, y1 + dy\*tmax)
4. If the line crosses over the window, you will see (x1 + dx\*tmin, y1 + dy\*tmin) and (x1 + dx\*tmax, y1 + dy\*tmax) are [intersection](https://www.cs.helsinki.fi/group/goa/viewing/leikkaus/intersec.html) between line and edge.

## Άσκηση 4 – Παράδειγμα Liang-Barsky

Chart, line chart

Description automatically generated

We calculate the 4 inequalities t\_i:

tB = (y0 – ymin) / -Δy = (0 – 1) / - (6 – 0) = 1/6

tL = (x0 – xmin) / -Δx = ( 0.5 – 1) / - (6 – 0.5) = - 0.5 / - 5.5 = 0.09

tR = (xmax – x0) / Δx = (5 – 0.5) / 5.5 = 0.81

tT = (ymax – y0) / Δy = (4 – 0 ) / 6 – 0 = 4/6

t\_min = 1/6

t\_max = 4/6

Draw the line: (x0 + dx\*tmin, y0 + dy\*tmin) to (x0 + dx\*tmax, y0 + dy\*tmax):

(0.5 + 5.5\*1/6, 0 + 6\*1/6) to (0.5 + 5.5\*4/6, 0 + 6\*4/6) = (1.416 , 1 ) , (4.16, 4)

## Άσκηση 2 - Cohen Algorithm

Step 1 : Assign a region code for two endpoints of given line.

Step 2 : If both endpoints have a region code 0000

then given line is completely inside.

Step 3 : Else, perform the logical AND operation for both region codes.

Step 3.1 : If the result is not 0000, then given line is completely

outside.

Step 3.2 : Else line is partially inside.

Step 3.2.1 : Choose an endpoint of the line

that is outside the given rectangle.

Step 3.2.2 : Find the intersection point of the

rectangular boundary (based on region code).

Step 3.2.3 : Replace endpoint with the intersection point

and update the region code.

Step 3.2.4 : Repeat step 2 until we find a clipped line either

trivially accepted or trivially rejected.

Step 4 : Repeat step 1 for other lines

## Άσκηση 3 – Παράδειγμα Cohen Algorithm

Chart, line chart

Description automatically generated

Τρέξιμο του Αλγόριθμου Cohen για το παραπάνω παράδειγμα:

Ορισμός αρχικών τιμών:

X\_min = 1, y\_min = 1, x\_max = 5, y\_max = 4 // ορισμός του παραλληλόγραμμου

X\_0 = 0.5, Y\_0 = 0, X\_1 = 6, Y\_1 = 6 // ορισμός ευθείας

# Ορισμός των τμημάτων αποκοπής

INSIDE **=** 0  # 0000

LEFT **=** 1    # 0001

RIGHT **=** 2   # 0010

BOTTOM **=** 4  # 0100

TOP **=** 8     # 1000

Αντιστοίχιση (Mapping) των σημείων P\_0, P\_1 στα αντίστοιχα τμήματα με τις πράξεις Bitwise OR

Code\_0 = 0101 (Left-Bottom)

Code\_1 = 1010 (Right-Top)

**While True:**

Έλεγχος αν η γραμμή είναι ολόκληρη εντός ή ολόκληρη εκτός

Διαλλέγουμε ένα άκρο που είναι εκτός (έστω το P\_0)

Υπολογίζουμε τα σημείο τομής με τους άξονες μέσω της φόρμουλας: έστω για το παράδειγμα ξεκινάμε από Bottom – Top – Left – Right:

y = y1 + slope \* (x - x1),

x = x1 + (1 / slope) \* (y - y1)

x = 0.5 + 5.5/6 \* 1 = 1.41

y = y\_min = 1

νέες συντεταγμένες: P0(1.41,1)

υπολόγισε το νέο Code\_0 …

Τρέχει και πάλι το loop για το 2ο σημείο P1, παρομοίως

x = x0 + (ytop – y1) / m = 1.41 + (4-1) \* 5.5/6 = 4.16

P1 = (4.16,4)

Ο αλγόριθμος τερματίζει αφού και τα δύο σημεία θα βρίσκονται εντός.

## Άσκηση 5 - Cohen – Southerland 3D Algorithm

O πίνακας των bits επεκτείνεται σε

Text, chat or text message

Description automatically generated

Χρησιμοποιούμε τις παραμετρικές εξισώσεις για τα σημεία τομής του ευθύγραμμου τμήματος

Text, letter

Description automatically generatedText

Description automatically generated

## Άσκηση 6 – Προγραμματιστική

*Estimated time 20hours*

Problems confronted: A lot of glut background details, Z-indexes, drawing callbacks.

Ο κώδικας βρίσκεται στο αρχείο ***sutherland-main.c*** και συνοδεύεται από το makefile.

O χρήστης, αφότου σημειώσει τις ακμές του πολυγώνου πατάει το πλήκτρο ‘d’ ή ‘D’ για να ζωγραφιστεί το πολύγωνο. Μετά, χρησιμοποιεί πατώντας το f1 τον «cropper» και όταν είναι έτοιμος πατάει το πλήκτρο ‘c’. Επίσης, όπως ζητήθηκε μπορεί να κρύβει τον cropper πατώντας το spacebar.

## Άσκηση 7 – Εφαρμoγή του Sutherland-Hodgman

The important formula for calculating the intersection points

Text, letter

Description automatically generated

Figure 2 <https://www.geeksforgeeks.org/polygon-clipping-sutherland-hodgman-algorithm-please-change-bmp-images-jpeg-png/?ref=gcse>

Diagram

Description automatically generated

A piece of paper with writing

Description automatically generated with medium confidence

## Άσκηση 9, RGB – HSV:

Divide r, g, b by 255

Compute cmax, cmin, difference

Hue calculation :

if cmax and cmin equal 0, then h = 0

if cmax equal r then compute h = (60 \* ((g – b) / diff) + 360) % 360

if cmax equal g then compute h = (60 \* ((b – r) / diff) + 120) % 360

if cmax equal b then compute h = (60 \* ((r – g) / diff) + 240) % 360

Saturation computation :

if cmax = 0, then s = 0

if cmax does not equal 0 then compute s = (diff/cmax)\*100

Value computation :

v = cmax\*100

HSV-RGB

C = (V/100)\*(S/100)

X = C\*(1 – | ((H/60)mod 2 )-1 |)

m = (V/100) – C

r = C,g = X,b = 0 if(0<= H <60)

r = X,g = C,b = 0 if(60<= H <120)

r = 0,g = C,b = X if(120<= H <180)

r = 0,g = X,b = C if(180<= H <240)

r = X,g = 0,b = C if(240<= H <300)

r = C,g = 0,b = X if(300<= H <360)

R = (r+m)\*255

G = (g+m)\*255

B = (b+m)\*255