

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΒΙΟΙΑΤΡΙΚΗ**

**1η ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ**

**ΓΙΑ ΤΟ ΕΡΓΑΣΤΗΡΙΟ ΤΟΥ ΜΑΘΗΜΑΤΟΣ:**

**ΓΡΑΦΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΘΕΜΑ:**

**ΥΛΟΠΟΙΗΣΗ ΕΝΟΣ ΑΛΓΟΡΙΘΜΟΥ ΓΕΜΙΣΜΑΤΟΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ ΔΙΕΠΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΕΦΑΡΜΟΓΩΝ ΤΗΣ OPENGL**

**ΕΚΠΟΝΗΘΗΚΕ ΑΠΟ ΤΟΥΣ ΦΟΙΤΗΤΕΣ:**

**ΚΩΝΣΤΑΝΤΙΝΙΔΗ ΠΑΥΛΟ, ΑΜ: 01651, 3o ΕΤΟΣ,**

**ΚΑΤΙΡΤΖΗ ΜΑΡΙΟ, ΑΜ: 01608, 3o ΕΤΟΣ**

**ΔΙΔΑΣΚΟΝΤΕΣ :**

**ΔΡΑΚΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ,**

**ΜΑΤΘΕΣ ΔΗΜΗΤΡΙΟΣ**

Λαμία, 2021

**ΠΕΡΙΕΧΟΜΕΝΑ**

* 1. **Περίληψη ....................................................................................................................... σελ. 3**
  2. **Abstract ….……………………………………………………………………………… p. 4**
  3. **Τεκμηρίωση Κώδικα .................................................................................................... σελ. 4**
  4. **Αναπαράσταση του Αποτελέσματός του Αλγορίθμου .............................................. σελ. 12**
  5. **Συμπεράσματα ............................................................................................................. σελ. 13**
  6. **Επίλογος ....................................................................................................................... σελ. 13**
  7. **Βιβλιογραφία ............................................................................................................... σελ. 14**
  8. **Ευρετήριο .................................................................................................................... σελ. 15**
  9. **Περίληψη**

Η εργασία που επωμιστήκαμε αφορά τη διαδικασία γεμίσματος ενός σχήματος με χρώμα. Ο αλγόριθμος που υλοποιήσαμε βασίζεται στον Αλγόριθμο Σάρωσης Πολυγώνου (Scanline Polygon Fill). Ο Αλγόριθμος Σάρωσης Πολυγώνου χρησιμοποιείται για να γεμίσει το εσωτερικό ενός πολυγώνου. Προκειμένου να βρεθούν τα εικονοστοιχεία που περικλείονται από το πολύγωνο, σημειώνονται τα σημεία τομής της εκάστοτε γραμμής σάρωσης, που σαρώνει το πολύγωνο από τη χαμηλότερη θέση (κατά y) έως την υψηλότερη, με το περίγραμμα του πολυγώνου (Κ.Μουστάκας κ.α, 2015, σελ. 40 ). Το πρόγραμμα υλοποιήθηκε σε γλώσσα προγραμματισμού C++ με τη χρήση της διεπαφής προγραμματισμού εφαρμογών της OpenGL1.

***Λέξεις Κλειδιά:*** *OpenGL, γέμισμα, Αλγόριθμος Σάρωσης Πολυγώνου, τομές, κορυφές, πλευρές*

Ο αλγόριθμος συνοψίζεται στα ακόλουθα βήματα:

**Βήμα 1.** Για κάθε θέση της γραμμής σάρωσης, εντόπισε τα σημεία τομής της γραμμής σάρωσης με τις ακμές του πολυγώνου.

**Βήμα 2.** Αποθήκευσε στους μονοδιάστατους πίνακες left και right, την τετμημένη της αριστερής και της δεξιάς τομής αντίστοιχα της γραμμής σάρωσης με το περίγραμμα του πολυγώνου.

**Βήμα 3.** Χρωμάτισε τα εικονοστοιχεία που βρίσκονται ανάμεσα σε ζεύγη σημείων τομής, δηλαδή για κάθε γραμμή σάρωσης (από κάτω προς τα πάνω), γέμισε με χρώμα τα εσωτερικά σημεία του πολυγώνου.

* Για να δουλέψει σωστά, ο αλγόριθμος προϋποθέτει από τον χρήστη να ορίσει τις συντεταγμένες των κορυφών του σχήματος και να καλέσει την συνάρτηση εύρεσης τομών, μεταξύ σημείων που όταν συνδέθούν μέσω ενός ευθυγράμμου τμήματος, θα προκύψει κάποια πλευρά του σχήματος.

*1 Ο αναγνώστης μπορεί να αναζητήσει την αναλυτικότερη επεξήγηση των συναρτήσεων στον ιστότοπο: <https://www.khronos.org/registry/OpenGL-Refpages/>*

* Υπάρχουν κάποια είδη πολυγώνων προκαθορισμένα, προκειμένου μέσω του μενού επιλογών, ο χρήστης να μπορεί άμεσα να δεί το αποτέλεσμα του αλγορίθμου.

**1.2 Abstract**

The work we have been assigned to do, concerns the process of filling a shape with color. The algorithm we implemented is based on the Scanline Polygon Fill Algorithm. The Polygon Scan Algorithm is used to fill the inside of a polygon. In order to find the pixels enclosed by the polygon, the intersection points of each scan line, which scans the polygon from the lowest position (by y) up to the highest, are marked with the outline of the polygon (K. Moustakas et al., 2015, p. 40). The program was implemented in C ++ programming language using the OpenGL1 application programming interface.

The algorithm is summarized in the following steps:

**Step 1.** For each position of the scan line, locate the intersections of the scan line with the edges of the polygon.

**Step 2.** Save in the one-dimensional arrays named: left and right, the intersection of the left and right sections respectively of the scan line with the outline of the polygon.

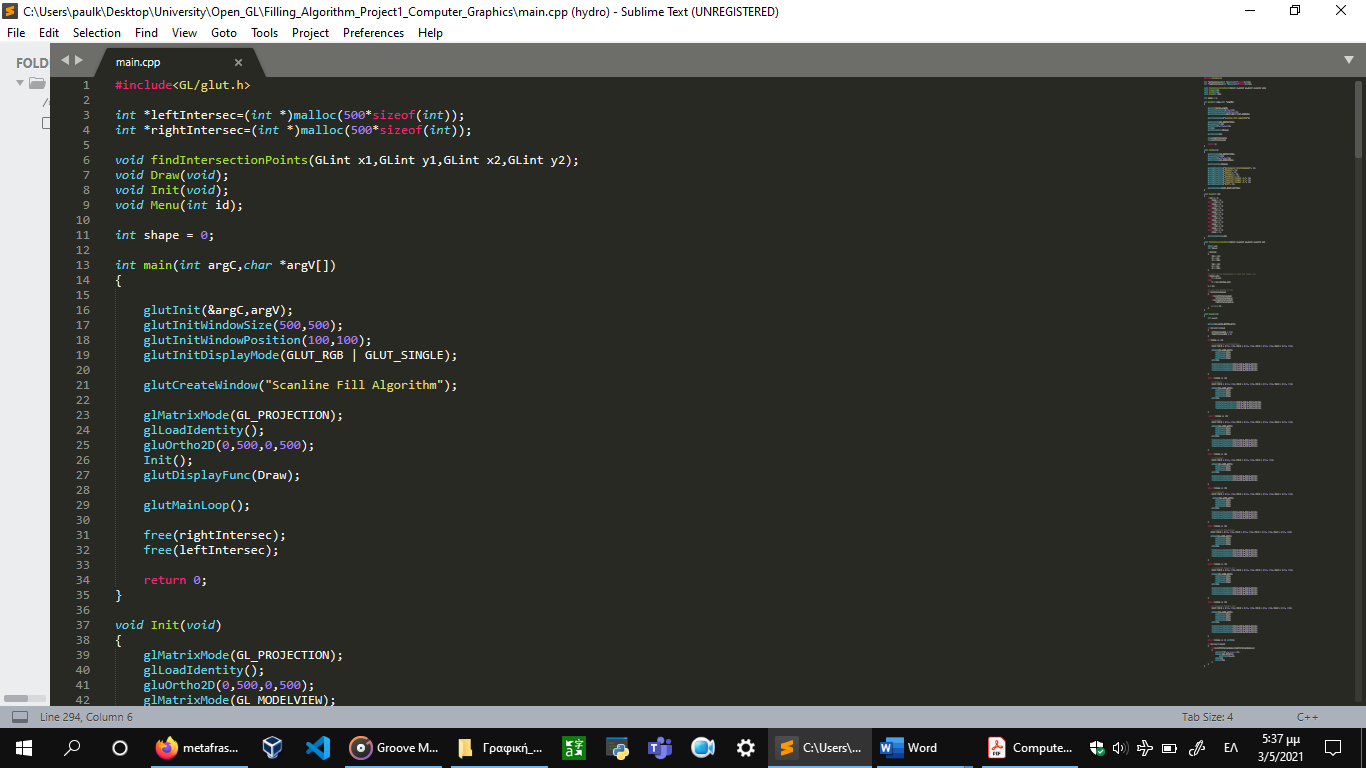
**Step 3.** Paint the pixels between pairs of intersection points, i.e., for each scan line (from bottom to top), fill the inner points of the polygon with color.

* To work properly, the algorithm requires the user to define the coordinates of the vertices of the shape and call the intersection finding function between points that when connected via a straight line will result in creating a side of the polygon.
* There are some types of polygons predefined so that through the option menu, the user can directly see the result of the algorithm.

*1 The reader can search for the more detailed explanation of the functions of the OpenGL API on the website:* [*https://www.khronos.org/registry/OpenGL-Refpages/*](https://www.khronos.org/registry/OpenGL-Refpages/)

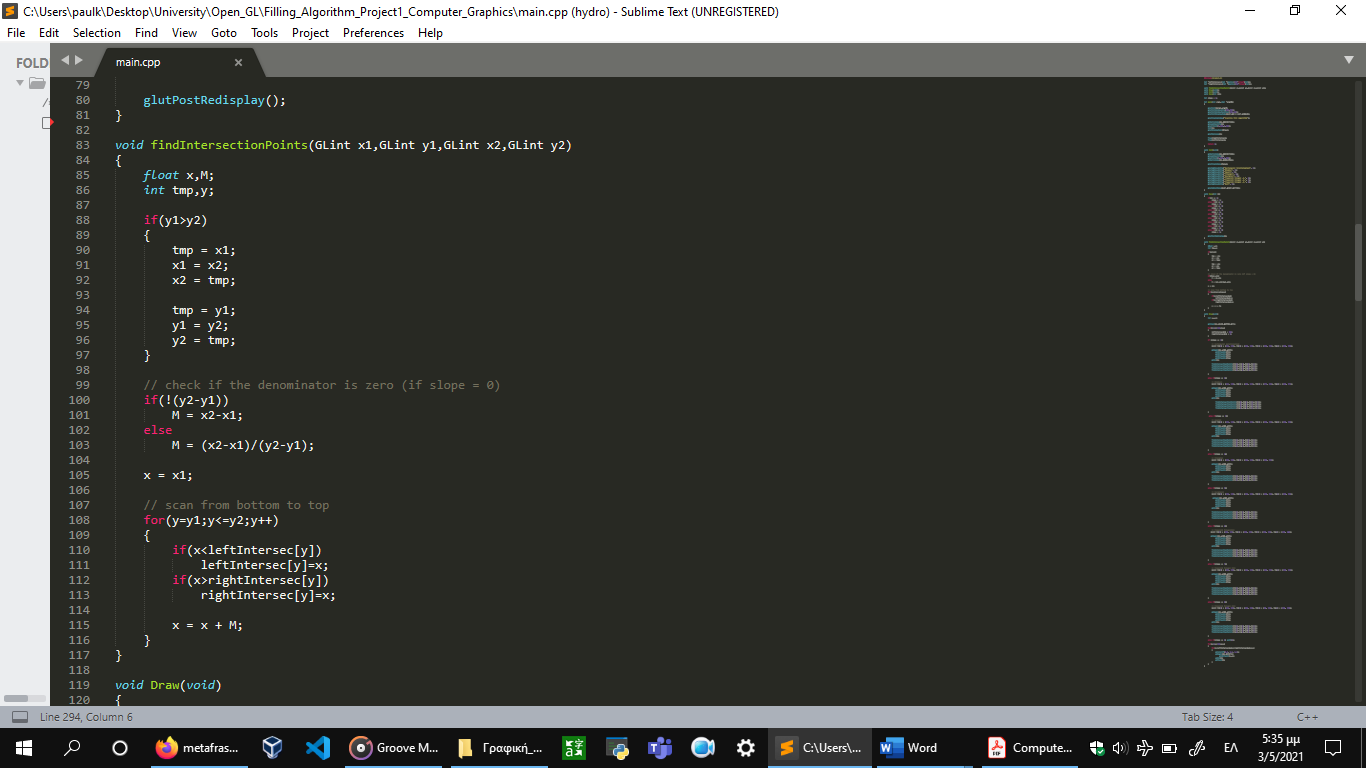
**1.3 Τεκμηρίωση Κώδικα**

Αρχικά, όπως φαίνεται στην παρακάτω εικόνα, χρησιμοποιούμε την εντολή του μικροεπεξεργαστή: #include <GL/glut.h>, για να μπορέσουμε να ενσωματώσουμε στον κώδικά μας χρήσιμες συναρτήσεις που παρέχονται από την Διεπαφή Προγραμματισμού Εφαρμογών (API) της OpenGL, για τον σχεδιασμό των σχημάτων. Έπειτα, αρχικοποιούμε δύο δυναμικούς πίνακες (leftIntersec και rightIntersec) με τη βοήθεια της συνάρτησης malloc(). Σε αυτούς τους πίνακες θα αποθηκεύονται οι τετμημένες των τομών του περιγράμματος του σχήματος με τις γραμμές σάρωσης(η αριστερή και η δεξιά τετμημένη για να είμαστε ακριβείς). Στη συνέχεια, ακολουθούν τα πρωτότυπα των δύο συναρτήσεων που χρησιμοποιούνται για το «γέμισμα» του πολυγώνου (findIntersectionPoints, Draw). Κατόπιν, ορίζονται τα πρωτότυπα των συναρτήσεων (Init, Menu) που μας επιτρέπουν να δημιουργήσουμε το μενού επιλογών.



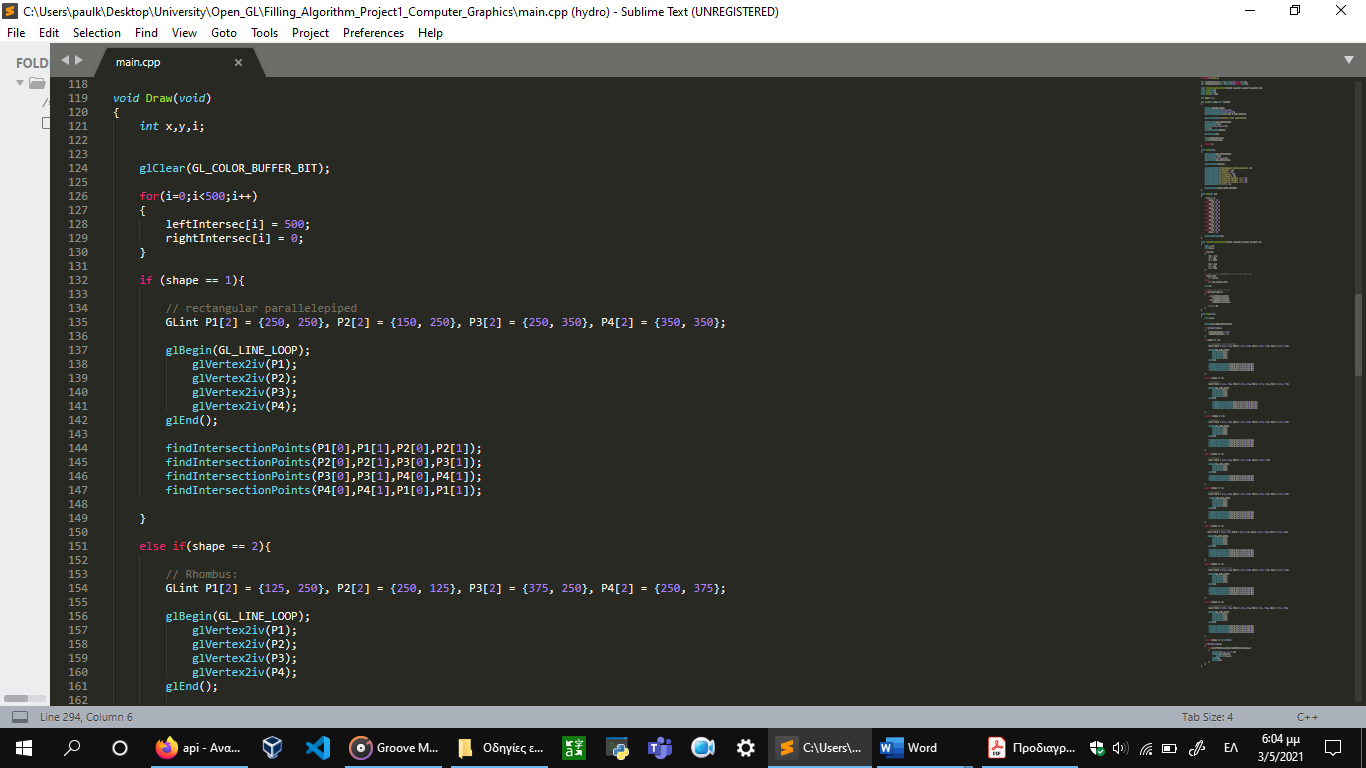
Εικόνα 1

Η συνάρτηση findIntersectionPoints δέχεται τέσσερα ορίσματα τύπου GLint, τα οποία αναπαριστούν τις συντεταγμένες δύο γειτονικών κορυφών του σχήματος. Στο εσωτερικό μπλοκ της συνάρτησης, ορίζουμε δύο μεταβλητές κινητής υποδιαστολής: την x και την M, και δύο μεταβλητές ακεραίου τύπου (int). Έπειτα, ελεγχουμε αν τα ορίσματα έχουν τοποθετηθεί στη σωστή σειρά (αν η συνάρτηση έχει κληθεί σωστά), και κάνουμε εναλλαγή τιμών έτσι ώστε το σημείο με τη μεγαλύτερη τεταγμένη να τοποθετηθεί στην κατάλληλη μεταβλητή. Κατόπιν, βρίσκουμε την κλίση του ευθυγράμμου τμήματος αφού πρώτα ελέγξουμε αν είναι μηδενική, για να εμποδίσουμε τη διαίρεση με το μηδέν. Στην προηγούμενη περίπτωση θέτουμε Μ = x2 - x1, (δηλαδή ως κλίση το 0), (όπως φαίνεται και στην εικόνα 2, γραμμή 101). Τέλος, στις γραμμές 108 – 116, μέσω μιας επαναληπτικής διαδικασίας, γεμίζουμε τους πίνακες leftIntersec και rightIntersec με τα σημεία τομής της εκάστοτε ευθείας σάρωσης με το πολύγωνο. Στο αρχικό σημείο του ευθυγράμμου τμήματος που αποτελεί κάποια πλευρά του σχήματος, προσθέτουμε την κλίση Μ, ώστε να λάβουμε το επόμενο σημείο του (γραμμή 115, εικόνα 2).



Εικόνα 2

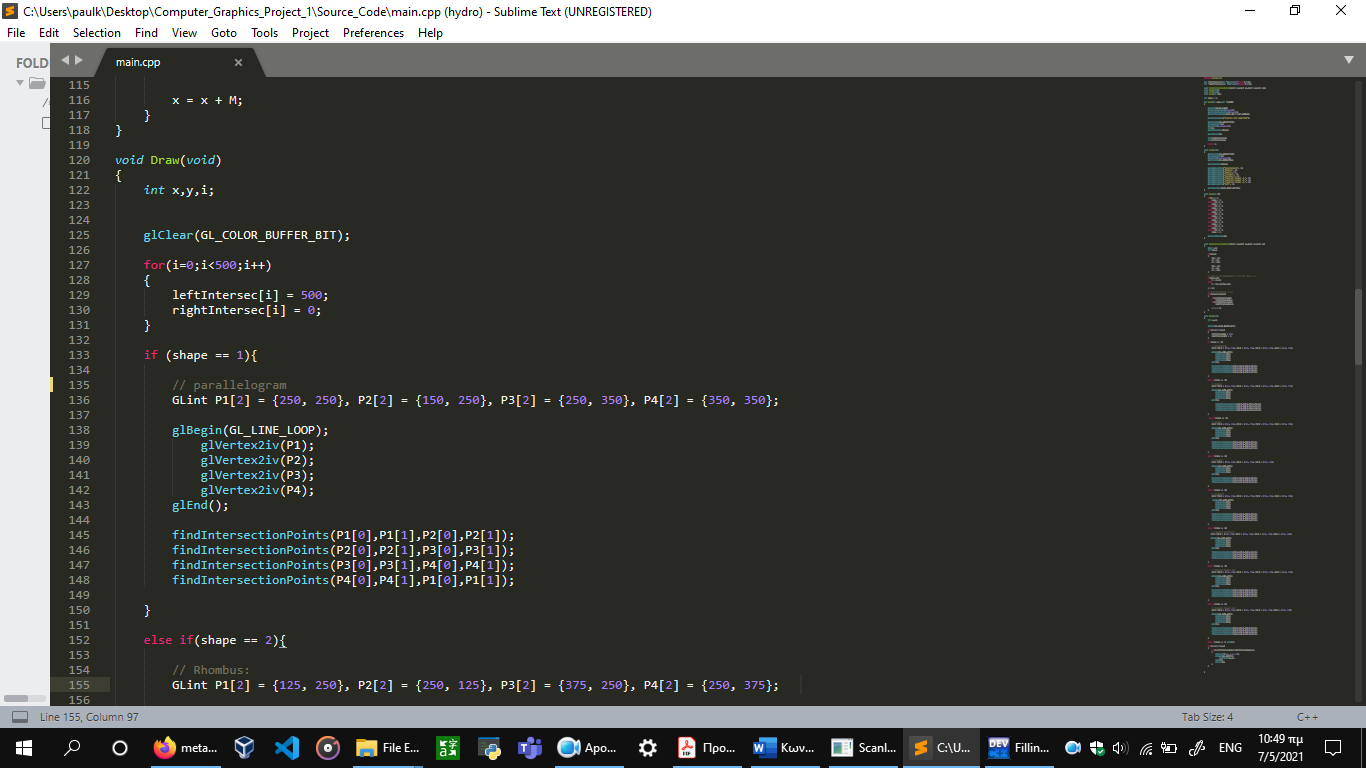
Στην συνάρτηση **Draw**, ορίζονται οι κορυφές (με σημεία) του σχήματος που πρόκειται να σχεδιαστεί, στην μορφή πινάκων τύπου Glint. Προτού σχεδιαστεί το περίγραμμα του πολυγώνου, είναι σημαντικό να καθαριστούν οι καταχωρητές βάθους και χρωμάτων μέσω της κλήσης της συνάρτησης: glClear(GL\_COLOR\_BUFFER\_BIT). Στη συνέχεια, αρχικοποιούμε τους πίνακες τομών με 500 και 0 αντίστοιχα, αριθμοί οι οποίοι αποτελούν τα όρια του παραθύρου θέασης.



Εικόνα 3

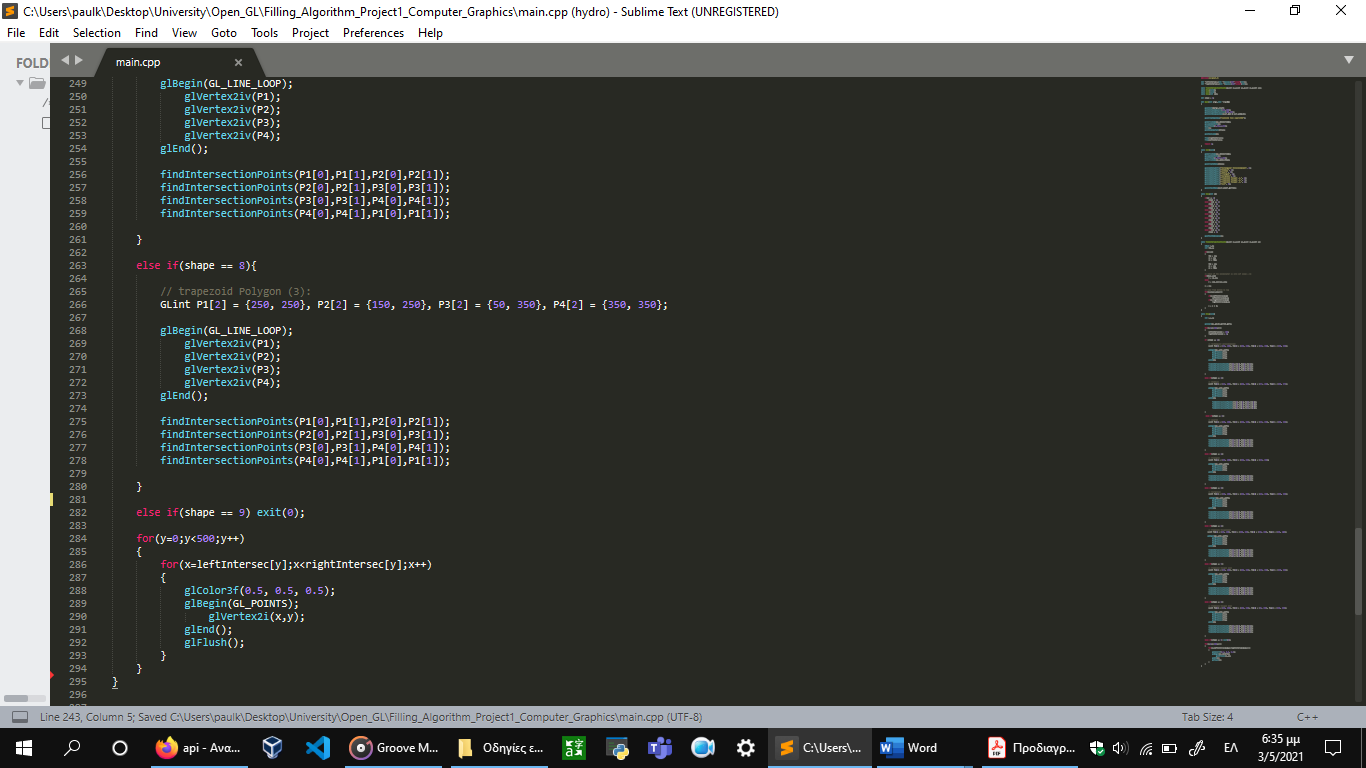
Ενδεικτικά, παρακάτω αναλύεται ο κώδικας που περιέχεται στην συνάρτηση Draw για τη σχεδίαση και το γέμισμα του παραλληλογράμμου. Ωστόσο, η διαδικασία είναι όμοια για την σχεδίαση και των υπολοίπων σχημάτων και για τον λόγο αυτό δεν θα αναλυθεί ο κώδικας των επόμενων σχημάτων. Η επιλογή της σχεδίασης των σχημάτων γίνεται με τη βοήθεια των δομών ελέγχου if-else και της καθολικής μεταβλητής shape.

Στις γραμμές 137 – 142 (Εικόνα 4), σχεδιάζεται το περίγραμμα του πολυγώνου με την βοηθεία των συναρτήσεων της βιβλιοθήκης glut.h. Πιο συγκεκριμένα, εμφωλευμένα μεταξύ της glBegin και glEnd, η κλήση της συναρτήσης glVertex2iv, μας επιτρέπει να σχεδιάσουμε κλειστά σχήματα μέσω ευθυγράμμων τμημάτων, των οποίων τα άκρα τους εισάγονται ως παράμετροι στη συνάρτηση. Έπειτα, καλούμε ανάλογα με το σχήμα τόσες φορές την συνάρτηση findIntersectionPoints όσες είναι και οι πλευρές του πολυγώνου (Είναι σημαντικό να κληθεί η συνάρτηση findIntersectionPoints με ορίσματα γειτονικές κορυφές).



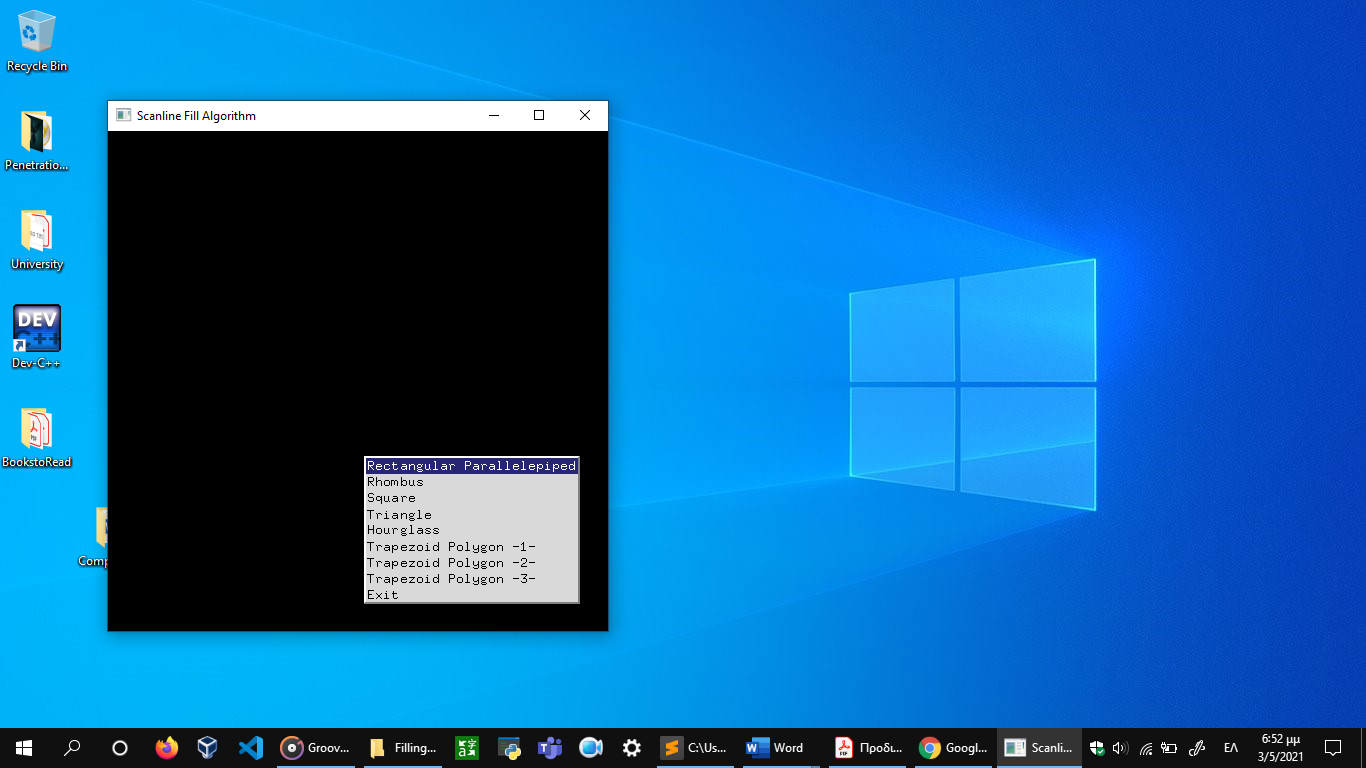
Εικόνα 4

Τέλος, με την επαναληπτική διαδικασία που φαίνεται στην εικόνα, «γεμίζουμε» το πολύγωνο από αριστερά προς τα δεξιά και από κάτω προς τα πάνω, χρωματίζοντας κάθε φορά ένα εσωτερικό σημείο του πολυγώνου με την χρήση της συνάρτησης glVertex2i(x,y), (γραμμή 290).



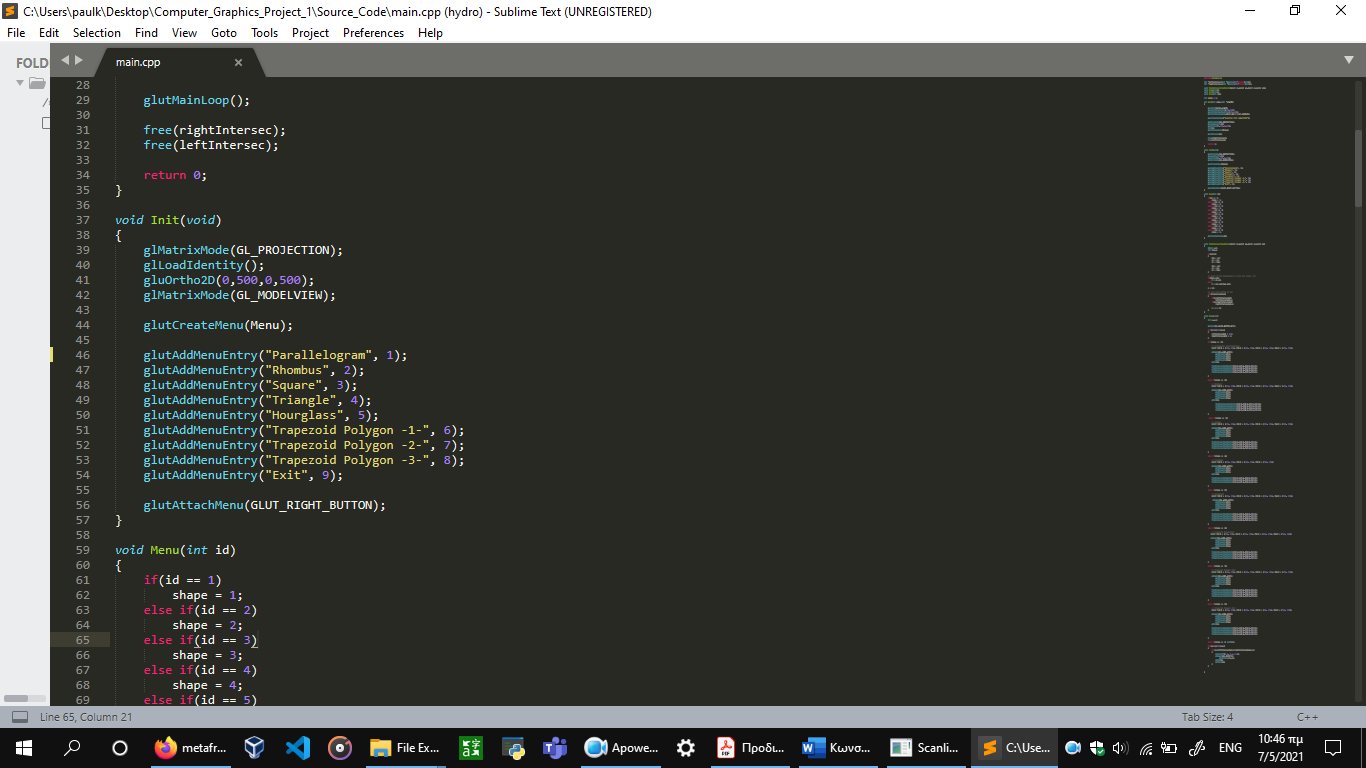
Εικόνα 5

Όπως αναφέρθηκε και προηγουμένως, ο χρήστης έχει την δυνατότητα να επιλέξει ανάμεσα σε 8 προκαθορισμένα σχήματα για να γεμίσει με χρώμα με τη βοήθεια του αλγορίθμου. Με το που τρέξει ο χρήστης το πρόγραμμα, εαν πατήσει το δεξί κλίκ του ποντικιού μέσα στο παράθυρο που εμφανίζεται, προβάλλεται το παρακάτω μενού επιλογών:

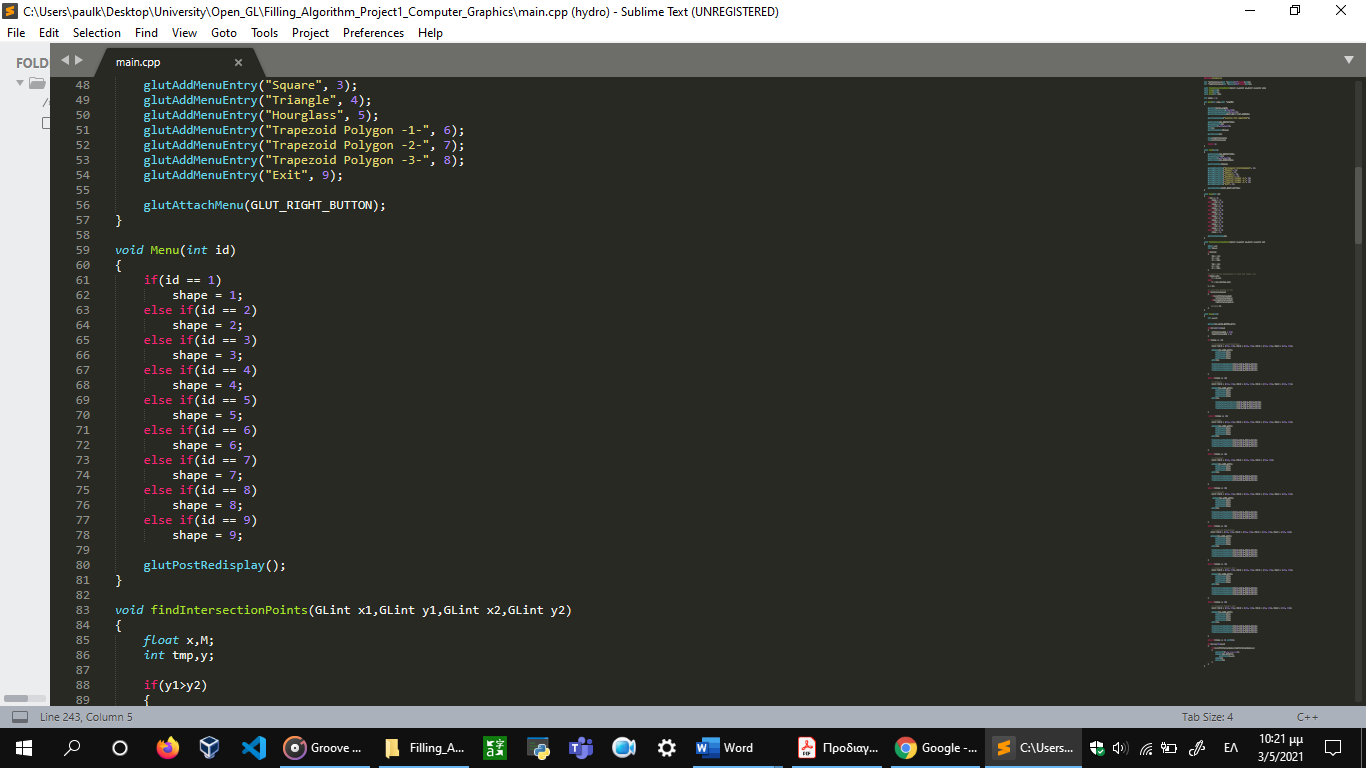


Εικόνα 6

Αυτό το μενού, ρυθμίζεται από τις συναρτήσεις Init και Menu, όπως φαίνονται στις εικόνες 7 και 8 αντίστοιχα. Στο εσωτερικό μπλόκ της συνάρτησης Init, δημιουργούμε το μενού με την συνάρτηση glutCreateMenu και προσθέτουμε τις διάφορες μπάρες επιλογής σχήματος με την συνάρτηση glutAddMenuEntry. Η συνάρτηση Menu, καθορίζει κάθε φορά ποιό σχήμα θα γεμίζεται, αποθηκεύοντας στη καθολική μεταβλητή shape την κατάλληλη τιμή. Μέσω της συνάρτησης glutPostRedisplay, ο χρήστης έχει τη δυνατότητα να γεμίσει κάποιο επόμενο σχήμα χωρίς να «ξανατρέξει» το πρόγραμμα.

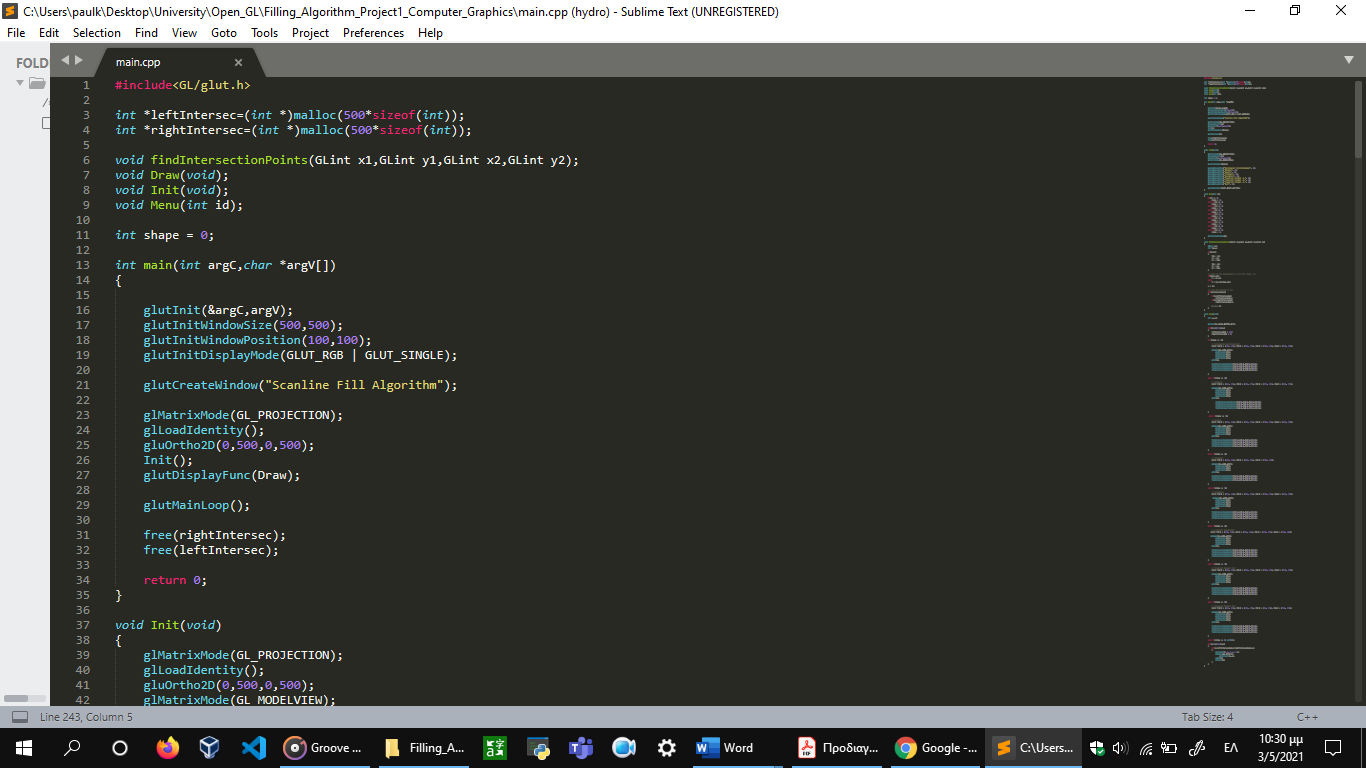


Εικόνα 7



Εικόνα 8

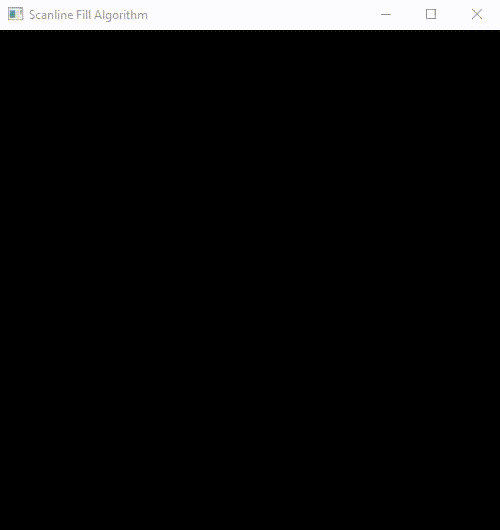
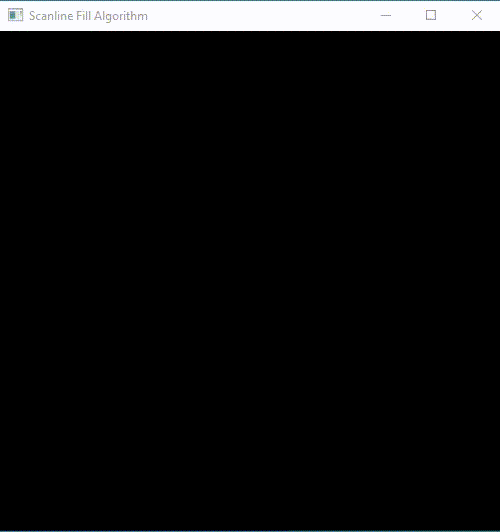
Με την συνάρτηση main, ορίζουμε το μέγεθος του παραθύρου θέασης και τη θέση του στην οθόνη με τις συναρτήσεις glutInitWindowSize(500,500) και glutInitWindowPosition(100,100) αντίστοιχα. Έπειτα, επιλέγουμε ως τη μορφή των χρωμάτων σχεδίασης το RGB και χρησιμοποιούμε μονό καταχωρητή σχεδίασης. Με τη συνάρτηση glutCreateWindow δημιουργούμε το παράθυρο προβολής, ενώ με τις συναρτήσεις των γραμμών 23 – 25 επιλέγουμε τον μετασχηματισμό προβολής, καθορίζουμε το παράθυρο θέασης και με την glutDisplayFunc καλούμε την συνάρτηση Draw. Όταν τελειώσει η εκτέλεση του προγράμματος, απελευθερώνουμε την μνήμη που δεσμέυτηκε από τους δυναμικούς πίνακες (γραμμές 31 – 32).



Εικόνα 9

**1.4 Αναπαράσταση του Αποτελέσματος του Αλγορίθμου**

Έστω λοιπόν ότι θέλουμε να γεμίσουμε ένα τετράγωνο και ένα παραλληλόγραμμο με χρώμα. Η διαδικασία είναι απλή. Αφού τρέξουμε το πρόγραμμα, πατάμε το δεξί κουμπί του ποντικιού και επιλέγουμε «Square» (τετράγωνο στα αγγλικά) για το τετράγωνο και «Parallelogram» για το παραλληλόγραμμο αντίστοιχα, όπως φαίνεται παρακάτω:



*Εικόνα 10 Εικόνα 11*

Ο χρήστης μπορεί να επιλέξει ακόμα 7 σχήματα για να γεμίσει ή μπορεί να τροποποιήσει τον αλγόριθμο(να προσθέσει τις συνεταγμένες των κορυφών του πολυγώνου) για να γεμίσει κάποιο σχήμα της αρεσκείας του.

**1.5 Συμπεράσματα**

Ο αλγόριθμος που παρουσιάστηκε προηγουμένως, δύναται να γεμίσει οποιοδήποτε σχήμα με χρώμα, αρκεί ο χρήστης να ορίσει σωστά τις συντεταγμένες των κορυφών του και να προσέξει αυτές να ορίζονται μέσα στα όρια του παραθύρου θέασης. Ακόμα, για πιο σύνθετα σχήματα (δηλαδή για σχήματα τα οποία κάποια γραμμή σάρωσης τέμνει σε περισότερα από δύο σημεία το περίγραμμά τους), είναι αναγκαίο να τροποποιηθούν οι πίνακες τομών, ώστε από μία διάσταση να έχουν δύο, για να συμπεριλάβουν όλες τις τομές. Ένας άλλος τρόπος θα ήταν να απλοποιήσει ο χρήστης είτε μόνος του είτε μέσω συνάρτησης το σύνθετο σχήμα σε τρίγωνα, καθώς γνωρίζουμε ότι τα τρίγωνα ως αρχέγονα, μπορούν να χρησιμοποιηθούν για να κατασκευάσουν οποιοδήποτε σχήμα, και έπειτα να καλέσει την συνάρτηση γεμίσματος σε κάθε ένα τρίγωνο ξεχωριστά.

**1.6 Επίλογος**

Εν κατακλείδι, η εργασία που ασχοληθήκαμε είχε ιδιαίτερο ενδιαφέρον για εμάς, λόγω του ότι μπορέσαμε να μάθουμε περισσότερα για την διεπαφή προγραμματισμού εφαρμογών της OpenGL και συγκεκριμένα το πώς γίνεται το γέμισμα ενός σχήματος με χρώμα. Οι γνώσεις που αποκτήσαμε με την ενασχόλησή μας με την εργασία αναμφίβολα θα αποτελέσουν έναυσμα για περαιτέρω μελέτη.

*[ Ο κώδικας του αλγορίθμου που αναπτύχθηκε: ]*

**1.7 Βιβλιογραφία**

1. Κ. Μουστάκας, Ι. Παλιόκας, Α. Τσακίρης, Δ.Τζοβάρας, “Γραφικά και Εικονική Πραγματικότητα”, 2015.

**1.8 Ευρετήριο**

Αλγόριθμος Σάρωσης Πολυγώνου, 3 rightIntersec, 3, 5

γέμισμα, 5, 7, 13 Square, 12

γεμίσματος, 3, 13 τετράγωνο, 12

C++, 3

Draw, 5, 6, 7, 11

findIntersectionPoints, 5, 7

glutAddMenuEntry, 9

glBegin, 7

glClear, 7

glCreateMenuEntry, 9

glEnd, 7

Glint, 5, 6

glutDisplayFunc, 11

glutInitWindowSize, 11

glutInitWindowPosition, 11

glutPostRedisplay, 9

glVertex2iv, 7, 8

#include <GL\glut.h>, 5

Init, 5, 9, 11

κορυφές/ων, 3, 5, 6, 7, 12, 13

leftIntersec, 3, 5

μονοδιάστατους πίνακες, 3

malloc, 5

Menu, 4, 5, 9

OpenGL, 3, 4, 5, 13

παραλληλόγραμμο, 12

πολυγώνου, 3, 4, 5, 6, 7, 8, 12

Parallelogram, 12