Γραφικά Υπολογιστών & Συστήματα Αλληλεπίδρασης
Αναφορά Προγραμματιστικής Άσκησης 1-Α:

Εξοικείωση με την χρήση βασικών βιβλιοθηκών γραφικών της OpenGL 3.3

Μέλη ομάδας:

Αντωνίου Χριστόδουλος AM: 2641
Τσιούρη Αγγελική AM: 3354

Α. Εισαγωγική σημείωση

Για την υλοποίηση της άσκησης αρχικά, μελετήσαμε τα δύο πρώτα εργαστήρια του μαθήματος για να εξοικειωθούμε σε θεωρητικό επίπεδο με τις βασικές έννοιες της OpenGL 3.3. Στη συνέχεια κάναμε προσθήκες και τροποποιήσεις στον ενδεικτικό κώδικα που μας δόθηκε για να πετύχουμε τα ζητούμενα του κάθε ερωτήματος. Η άσκηση έχει αναπτυχθεί σε Windows 10 με Visual Studio 2019. Το παραδοτέο .zip αρχείο περιλαμβάνει το αρχείο "Main.cpp" με τον κώδικα της άσκησης και τα συμπληρωματικά αρχεία "SimpleFragmentShader.fragmentshader" και "SimpleVertexShader.vertexshader" με τους shaders. Δεν συμπεριελαμβάνονται αρχεία των πακέτων ή του Visual Studio. Για να τρέξει το πρόγραμμα θα χρειαστεί να γίνει import σε ένα Visual Studio Project το οποίο περιέχει τις βιβλιοθήκες GLM, GLEW και GLFW. Ακολουθεί η αναλυτική επεξήγηση για την λειτουργία του προγράμματος χωρισμένη ανά ερώτημα.

Β. Ερώτημα (i)

Στο ερώτημα αυτό χρησιμοποιούμε την βιβλιοθήκη **GLFW** για την δημιουργία ενός παραθύρου. Στην αρχή του προγράμματος, κάνουμε include τον header της βιβλιοθήκης και δηλώνουμε μία μεταβλητή τύπου **GLFWwindow*** για το παράθυρο μας. Έπειτα, στην αρχή της main, αρχικοποιούμε την GLFW καλώντας την συνάρτηση **glfwInit()** και μετά καλούμε την **glfwCreateWindow(...)** για να δημιουργήσουμε το παράθυρο. Οι πρώτες δύο παράμετροι της συνάρτησης καθορίζουν το πλάτος και το ύψος του παραθύρου σε pixels αντίστοιχα. Επομένως, εδώ καθορίζουμε το επιθυμητό 800x800. Η τρίτη παράμετρος είναι ένα string με το όνομα του παραθύρου, "2Δ Σχήματα". Σημειώνεται πως λόγω encoding υπάρχει πιθανότητα οι ελληνικοί χαρακτήρες να μην εμφανίζονται σωστά. Για το background του παραθύρου καλούμε την συνάρτηση **glClearColor(...)**, με παραμέτρους: red=0.5, green=0, blue=0.5, alpha=1, οι οποίες αντιστοιχούν στο χρώμα dark purple χωρίς διαφάνεια. Για τον τερματισμό του προγράμματος με το πλήκτρο **n**, καλούμε την **glfwSetInputMode(...)** η οποία μας επιτρέπει να δεχτούμε input από το πληκτρολόγιο. Και έπειτα, τροποποιήσαμε την συνθήκη του do-while loop όπου γίνεται το render. Συγκεκριμένα χρησιμοποιούμε την συνάρτηση **glfwGetKey(...)**, με παραμέτρους το παράθυρο που δημιουργήσαμε και την σταθερά GLFW_KEY_N που αντιστοιχεί στο πλήκτρο n, για να ελέγξουμε εάν το πλήκτρο πατήθηκε. Στην περίπτωση αυτή η επανάληψη σταματά και το πρόγραμμα τερματίζει.

Γ. Ερώτημα (ii)

Το πρώτο πράγμα που χρειαζόμαστε για να ζωγραφίσουμε το ζητούμενο σχήμα είναι να δημιουργήσουμε ένα **Vertex Array Object**. Έπειτα, καθορίζουμε τις συντεταγμένες στην οθόνη. Θέλουμε να ζωγραφίσουμε 4 τρίγωνα, με κάθε τρίγωνο να αποτελείται από 3 σημεία (**vertices**) και με κοινό σημείο την αρχή των αξόνων (0, 0, 0) η οποία βρίσκεται στο κέντρο του

παραθύρου. Κάθε σημείο αποτελείται απο 3 συντεταγμένες **x**, **y**, **z** αντίστοιχα, με την τελευταία να είναι πάντα μηδενική καθώς τα τρίγωνα του ζητούμενου σχήματος είναι διδιάστατα. Με βάση τα παραπάνω, καθορίζουμε τις συντεταγμένες των μοντέλων μας σε ένα πίνακα **g_vertex_buffer_data** 36 θέσεων. (βλ. Ερώτημα (iv) για εικόνες υπολογισμού συντεταγμένων) Στη συνέχεια, πρέπει να "περάσουμε" τα τρίγωνα που καθορίσαμε στην OpenGL. Για τον λόγο αυτό δημιουργούμε έναν buffer τύπου **GLuint** και καλούμε τις συναρτήσεις **glGenBuffers(…)**, **glBindBuffer(…)** και **glBufferData(…)**. Σημειώνεται πως για να χρωματίσουμε τα τρίγωνα, έχουμε επέμβει στο αρχείο **SimpleFragmentShader.fragmentshader** το οποίο φορτώνεται στο πρόγραμμα μας μέσω της συνάρτησης **LoadShaders(…)**. Στη συνέχεια, ζωγραφίζουμε τα τρίγωνα μέσα στο do-while loop ώς εξής:

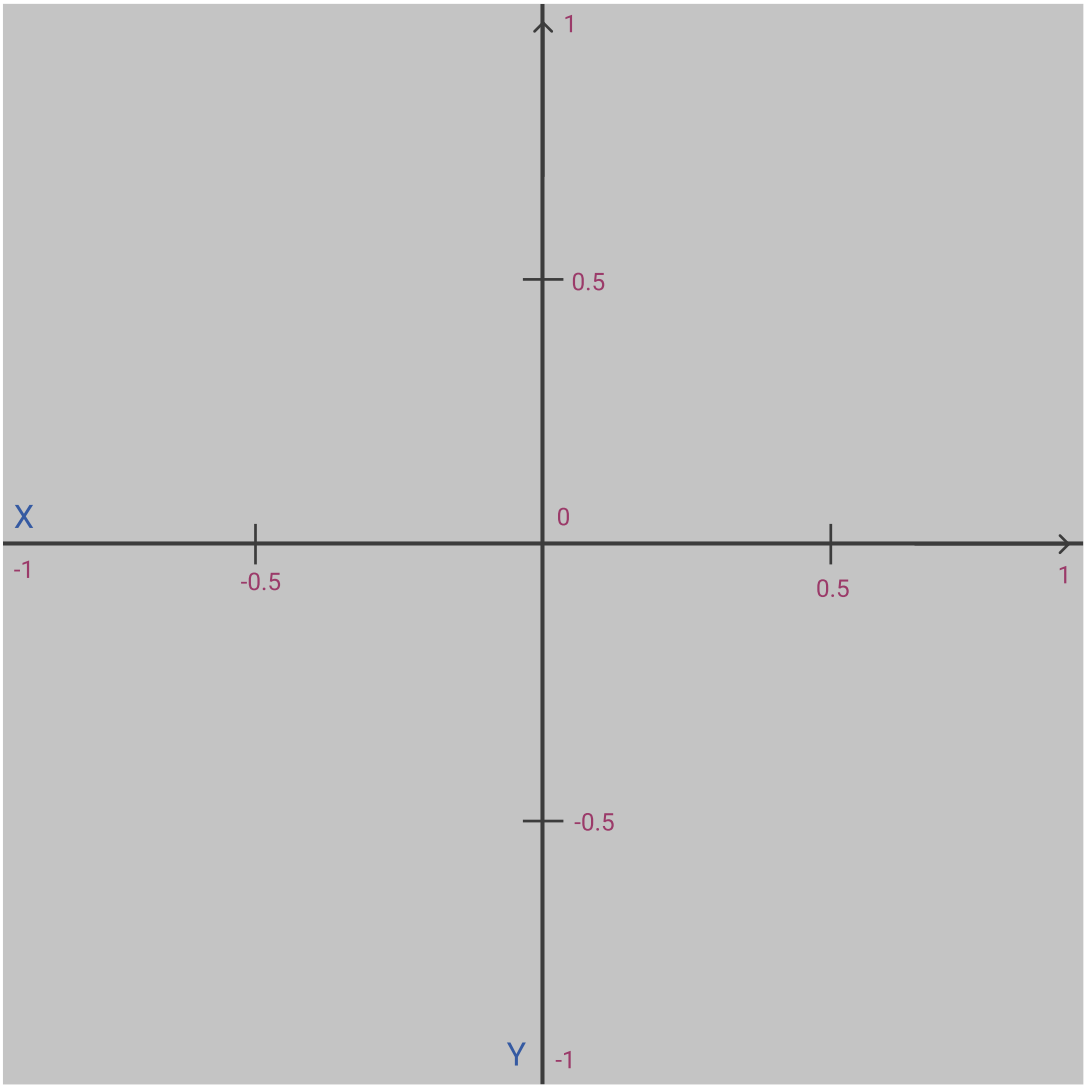
1) Κάνουμε clear το παράθυρο από τυχόν προηγούμενα σχέδια καλώντας την **glClear(…)**
2) Χρησιμοποιούμε τον shader για τον χρωματισμό των τριγώνων καλώντας την **glUseProgram(…)**
3) Χρησιμοποιούμε τον buffer που δημιουργήσαμε παραπάνω καλώντας τις συναρτήσεις **glEnableVertexAttribArray(…)**, **glBindBuffer(…)** και **glVertexAttribPointer(…)** καθορίζοντας πληροφορίες όπως είναι το μέγεθος ενός τριγώνου
4) Ζωγραφίζουμε τα τρίγωνα με την συνάρτηση **glDrawArrays(…)** όπου καθορίζουμε και τον αριθμό των vertices στον πίνακα μας που θέλουμε να ζωγραφιστούν. Στην περίπτωση μας έχουμε 4 τρίγωνα με το καθένα να αποτελείται απο 3 σημεία. Επομένως ξεκινάμε από το 0 μέχρι και το 12 που είναι το τελευταίο σημείο.
5) Τέλος, κάνουμε swap τους buffers με την συνάρτηση **glfwSwapBuffers(…)** έτσι ώστε να εμφανιστεί το σχέδιο στο παράθυρο μας.

Δ. Ερώτημα (iii)

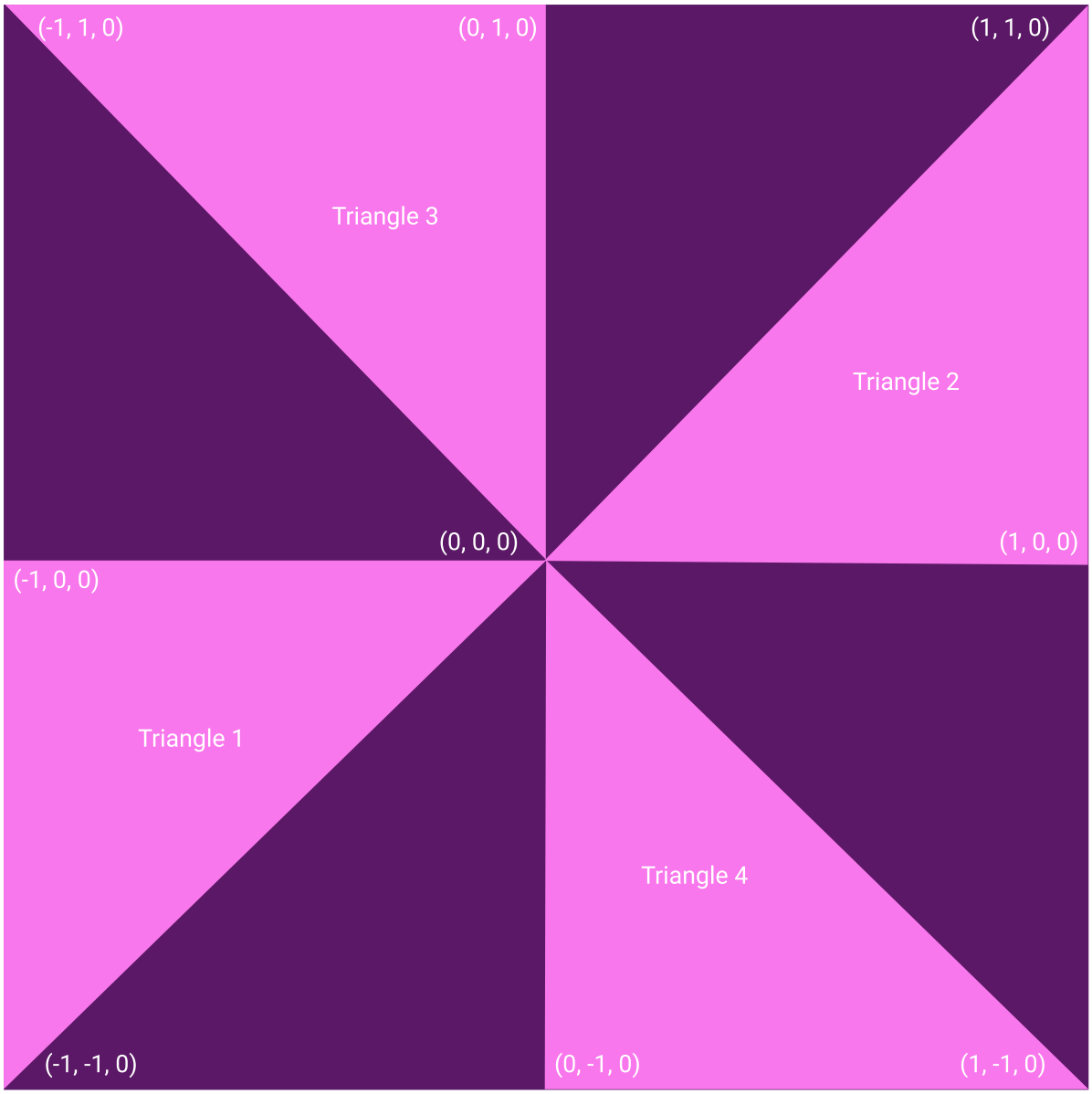
Το ερώτημα αυτό χωρίζεται σε δύο μέρη. Το πρώτο είναι να ανιχνέυσουμε πότε το πλήκτρο c πατιέται ή είναι πατημένο και το δεύτερο είναι να ζωγραφίσουμε το διαφορετικό σχήμα. Για το πρώτο χρησιμοποιούμε πάλι την συνάρτηση **glfwGetKey(…)** με παραμέτρους το παράθυρο που δημιουργήσαμε και την σταθερά **GLFW_KEY_C** που αντιστοιχεί στο πλήκτρο **c**. Επιπλέον, χρειαστήκαμε και μία flag μεταβλητή **c_press_flag** η οποία έχει την τιμή 1 όσο το πλήκτρο c παραμένει πατημένο, διαφορετικά έχει την τιμή 0, έτσι ώστε να μπορούμε να επαναφέρουμε το αρχικό σχέδιο όταν ο χρήστης αφήσει το πλήκτρο. Ο έλεγχος πραγματοποιείται στην αρχή του do-while loop. Στη σημείο αυτό παρατηρήσαμε πως το δεύτερο σχήμα δεν διαφέρει κατά πολύ από το αρχικό. Στα δεξία και αριστερά τρίγωνα, αλλάζουν δύο σημεία, ενώ στα πάνω και κάτω ένα σημείο. Έτσι, αποφασίσαμε να αλλάξουμε τα σημεία αυτά στον υπάρχον πίνακα που περιγράφει το μοντέλο του πρώτου σχεδίου. Για να δουλέψει η αλλαγή αυτή, χρειάστηκε να "περάσουμε" πάλι τον ανανεωμένο πίνακα στην OpenGL καλώντας την συνάρτηση **glBufferData(…)**. Φυσικά, η αντίστοιχη λειτουργία γίνεται και όταν ο χρήστης αφήνει το πλήκτρο c για να επαναφέρουμε τα τρίγωνα όπως είναι στο πρώτο σχήμα.

Ε. Ερώτημα (iv)

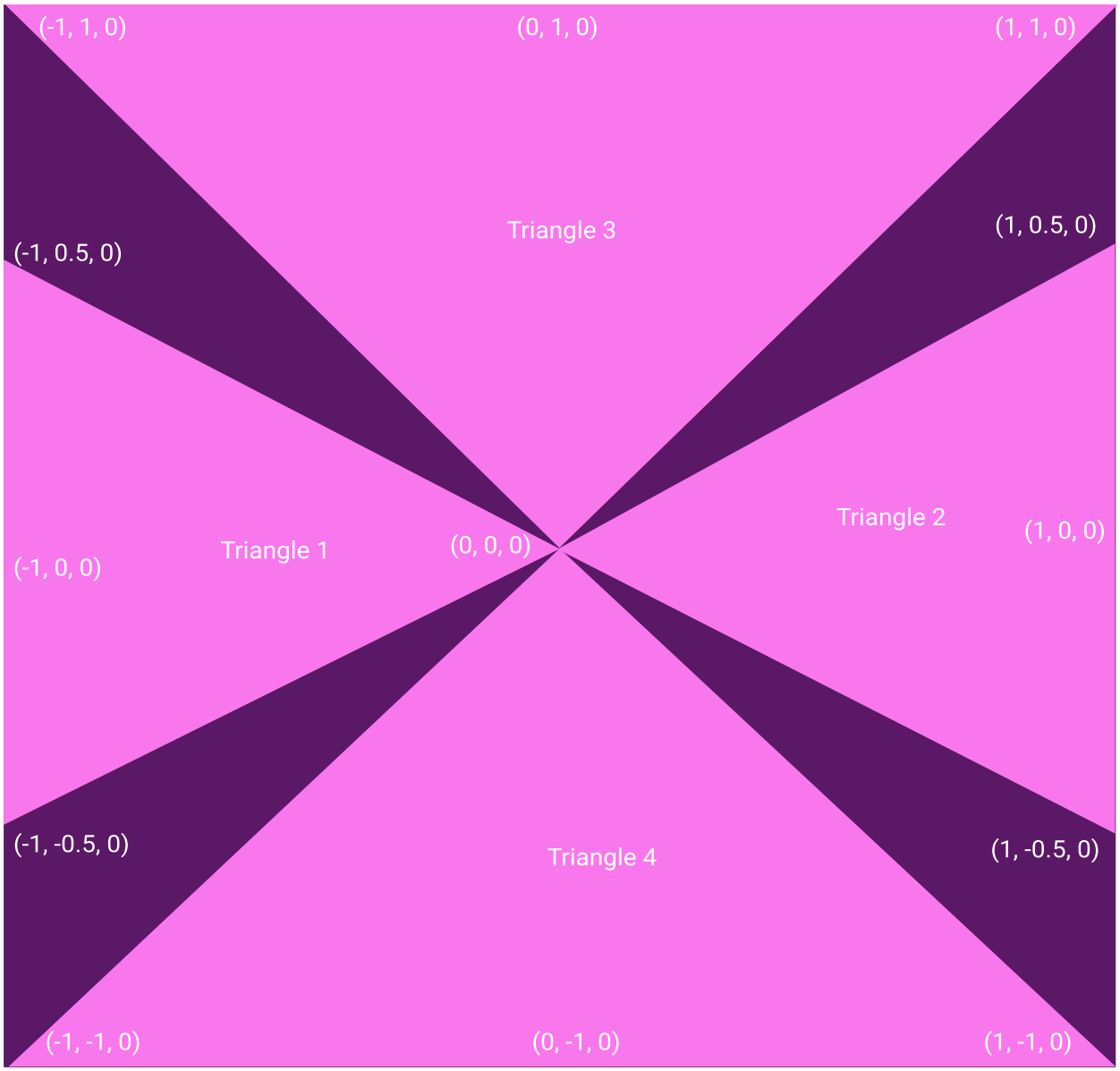
Το ερώτημα αυτό αναφέρεται στην δημιουργία της παρούσας αναφοράς καθώς και στον υπολογισμό των συντεταγμένων των τριγώνων των δύο σχημάτων. Ο υπολογισμός έχει γίνει θεωρόντας έναν διδιάστατο σύστημα αξόνων (x,y) με κέντρο το 0,0 ως κοινό σημείο μεταξύ των τριγώνων. Ακολουθούν στιγμιότυπα αναπαράστασης του συστήματος αξόνων και των τριγώνων με τον υπολογισμό των συντεταγμένων τους. Οι αναπαραστάσεις έχουν γίνει χειρονακτικά στο design tool Figma. Στις αναπαραστάσεις των σχεδίων συμπεριελαμβάνεται και ο τρίτος άξονας z, ο οποίος ωστόσο είναι μηδενικός καθώς το σχέδια είναι διδιάστατα.



Σύστημα Αξόνων


(-1, 1, 0)
(0, 1, 0)
(1, 1, 0)
Triangle 3
Triangle 2
(0, 0, 0)
(1, 0, 0)
(-1, 0, 0)
Triangle 1
Triangle 4
(-1, -1, 0)
(0, -1, 0)
(1, -1, 0)

Σχήμα (α)


(-1, 1, 0)
(0, 1, 0)
(1, 1, 0)
Triangle 3
(-1, 0.5, 0)
(1, 0.5, 0)
Triangle 2
(1, 0, 0)
Triangle 1
(-1, 0, 0)
(0, 0, 0)
(-1, -0.5, 0)
Triangle 4
(1, -0.5, 0)
(-1, -1, 0)
(0, -1, 0)
(1, -1, 0)

Σχήμα (β)