



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ**

---

# **ΔΙΔΙΑΣΤΑΤΗ ΣΚΗΝΗ**

ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΓΡΑΦΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΚΠΟΝΗΘΗΚΕ ΑΠΟ ΤΟΥΣ:

**Στεφανάτου Νεφέλη**

**Λάζαρο Κων/νο-Παναγιώτη**

ΔΙΔΑΣΚΩΝ ΚΑΘΗΓΗΤΗΣ:

**Δρακόπουλος Βασίλειος**

---

**ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ: 2020-2021**

# Περιεχόμενα

1	Εισαγωγή	2
2	Χάραξη διδιάστατης σκηνής	3
3	Χάραξη διδιάστατης σκηνής σε τριπλάσια ευκρίνεια	4
4	Μεταδιύθνηση με γκαουσιανό πθμό συνέλιξης	6
5	Μεταδιύθνηση με πθμό με όλα τα βάρη ίσα	7
6	Συμπεράσματα	8
7	Επίλογος	8

## Περίληψη

Η εργασία αυτή αναφέρεται στην υλοποίηση μιας διδιάστατης σκηνής με κώδικα καθώς και στην σύγκριση των αποτελεσμάτων διαφορετικών συνελίξεων πάνω σε αυτήν. Ο κώδικας αναπτύχθηκε σε γλώσσα C με χρήση συναρτήσεων της βιβλιοθήκης γραφικών OpenGL.

Το πρώτο κεφάλαιο αποτελεί μια συνοπτική ανάλυση του πηγαίου κώδικα που υλοποιήθηκε για τον σχεδιασμό της διδιάστατης σκηνής.

Στο δεύτερο κεφάλαιο περιγράφεται η υλοποίηση των συνελίξεων της σκηνής, αρχικά με έναν ηθμό συνέλιξης  $3 \times 3$  και στην συνέχεια με έναν ηθμό  $3 \times 3$  που έχει όλα τα βάρη του ίσα.

Έπειτα ακολουθεί η σύγκριση των διαφορετικών αποτελεσμάτων των δύο συνελίξεων.

Τέλος γίνεται διεξαγωγή συμπερασμάτων και καταγράφονται οι πηγές που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας.

## Abstract

This paper presents the implementation of a simple two dimensional scene. It also includes a comparison of the results that occur by applying two different convolution filters to the scene. The code was developed in C using functions from the OpenGL graphics library.

The first chapter is a brief analysis of the source code implemented for drawing the two-dimensional scene.

The second chapter describes the implementation of the two convolutions. The first one is implemented using a  $3 \times 3$  convolution filter and the second one is implemented using a  $3 \times 3$  filter with equal weights.

Then follows the comparison of the different results of the two convolutions.

Finally, conclusions are drawn and the sources used to implement the work are recorded.

Λέξεις κλειδιά: μεταδιύθιση [1], ηθμός, συνέλιξη [4], glsl [2], glfw [3]

## 1 Εισαγωγή

Με τον όρο διδιάστατα γραφικά εννοούμε την δημιουργία ψηφιακών εικόνων από διδιάστατα μοντέλα σε υπολογιστή. Σκοπός αυτής της εργασίας είναι η περιγραφή της χάραξης μιας απλής διδιάστατης σκηνής η οποία αποτελείται από ένα σύνολο τριγώνων. Με βάση τα ζητούμενα, η σκηνή θα χαραχθεί σε τριπλάσια ευκρίνεια και στη συνέχεια θα υποστεί επεξεργασία με χρήση δύο ηθμών. Ο πρώτος είναι ο ηθμός συνέλιξης  $3 \times 3$  και

ο δεύτερος είναι ένας πημός 3x3 ο οποίος έχει όλα του τα βάρη ίσα. Τα βάρη αμφότερων πημών είναι κανονικοποιημένα. Ακολουθεί αναλυτική περιγραφή της διαδικασίας καθώς και σύγκριση των διαφορετικών αποτελεσμάτων.

## 2 Χάραξη διδιάστατης σκηνής

Αρχικά, συμπεριλαμβάνονται (με `include`) οι βιβλιοθήκες που είναι απαραίτητες για την χάραξη της διδιάστατης σκηνής. Στην συνέχεια αρχικοποιούμε τις συντεταγμένες και τα χρώματα που θα έχουν οι κορυφές των τριγώνων της σκηνής. Η κάθε κορυφή περιγράφεται από 2 διανύσματα. Το ένα είναι το διάνυσμα θέσης ( $x,y$ ) και το άλλο είναι το διάνυσμα χρώματος RGB.

Με την συνάρτηση `error_callback` γίνεται έλεγχος για πιθανά λάθη κατά τη διάρκεια εκτέλεσης του προγράμματος. Σε περίπτωση που υπάρξει κάποιο σφάλμα, θα τυπωθεί στην οθόνη αντίστοιχο μήνυμα. Με την συνάρτηση `key_callback` γίνεται έλεγχος για το αν ο χρήστης έχει πατήσει κάποιο πλήκτρο (αν ο χρήστης πατήσει το πλήκτρο `escape` τότε θα κλείσει το παράθυρο της σκηνής). Η εργασία αναπτύχθηκε σε λειτουργικό σύστημα `linux`.

Στην κύρια συνάρτηση του προγράμματος (`main`) αρχικοποιείται η βιβλιοθήκη `glfw` (ελαφριά βοηθητική βιβλιοθήκη για χρήση με `OpenGL`) και δημιουργείται ένα παράθυρο λειτουργίας με ορίσματα το πλάτος και το ύψος της επιφάνειας σχεδίασης της εφαρμογής σε pixels, στο οποίο θα εμφανιστεί η διδιάστατη σκηνή.

Στην διδιάστατη σκηνή απεικονίζεται ένα ρομπότ το οποίο είναι σχεδιασμένο με βάση την τεχνοτροπία του κινέζικου παιχνιδιού `tangram`. Χρησιμοποιώντας την εντολή `glfwSwapInterval` σχεδιάζεται πρώτα το τρέχον πλαίσιο (`frame`) πριν αρχίζει να σχεδιάζεται το επόμενο. Οι σκιαστές (`shaders`) συνδέουν τον κώδικα με τα γραφικά. Για να χρησιμοποιήσουμε τους `shaders`, απλά τους καλούμε καθώς είναι γραμμένοι σε μια ειδική γλώσσα που λέγεται `glsl` και βρίσκονται πάνω από την κύρια συνάρτηση (πηγαίο αρχείο `main.c`, γραμμές 114-171). Υπάρχουν δύο τύποι `shader`:

1. Ο `vertex shader`. Μετατρέπει τις συντεταγμένες των κορυφών κάθε αρχηγόνου / σχήματος (στην περίπτωσή μας τρίγωνα) , σε συντεταγμένες σχεδίασης.
2. Ο `fragment shader`. Υπολογίζει το χρώμα και άλλα χαρακτηριστικά κάθε "θραύσματος" της σκηνής.

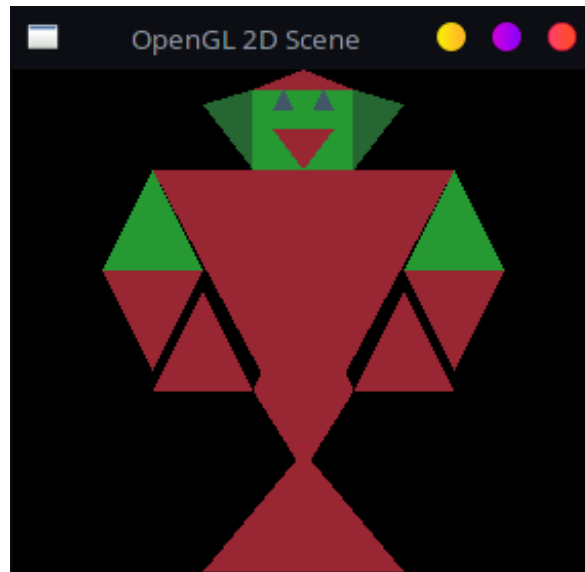
Με τον όρο `θραύσμα` εννοούμε την μονάδα απόδοσης που επηρεάζει το πολύ ένα μεμονωμένο εικονοστοιχείο (`pixel`) εξόδου. Το πρόσωπο του ρομπότ είναι τετράγωνο. Για να φτιαχτεί, ενώνονται τρία τρίγωνα μεταξύ τους με κατάλληλο τρόπο.

Με την εντολή `glViewport` ορίζεται η θύρα θέασης μέσα στο παράθυρο λειτουργίας. Εκεί θα σχεδιαστεί τελικά η διδιάστατη σκηνή.

Χρησιμοποιείται η εντολή `glClear(GL_COLOR_BUFFER_BIT)` έτσι ώστε να καθαριστεί ο ρυθμιστής (buffer) χρώματος.

Με την εντολή `glfwSwapBuffers(window)` γίνεται εναλλαγή των ρυθμιστών πλαισίου (πρόσθιο και οπίσθιο frame buffer).

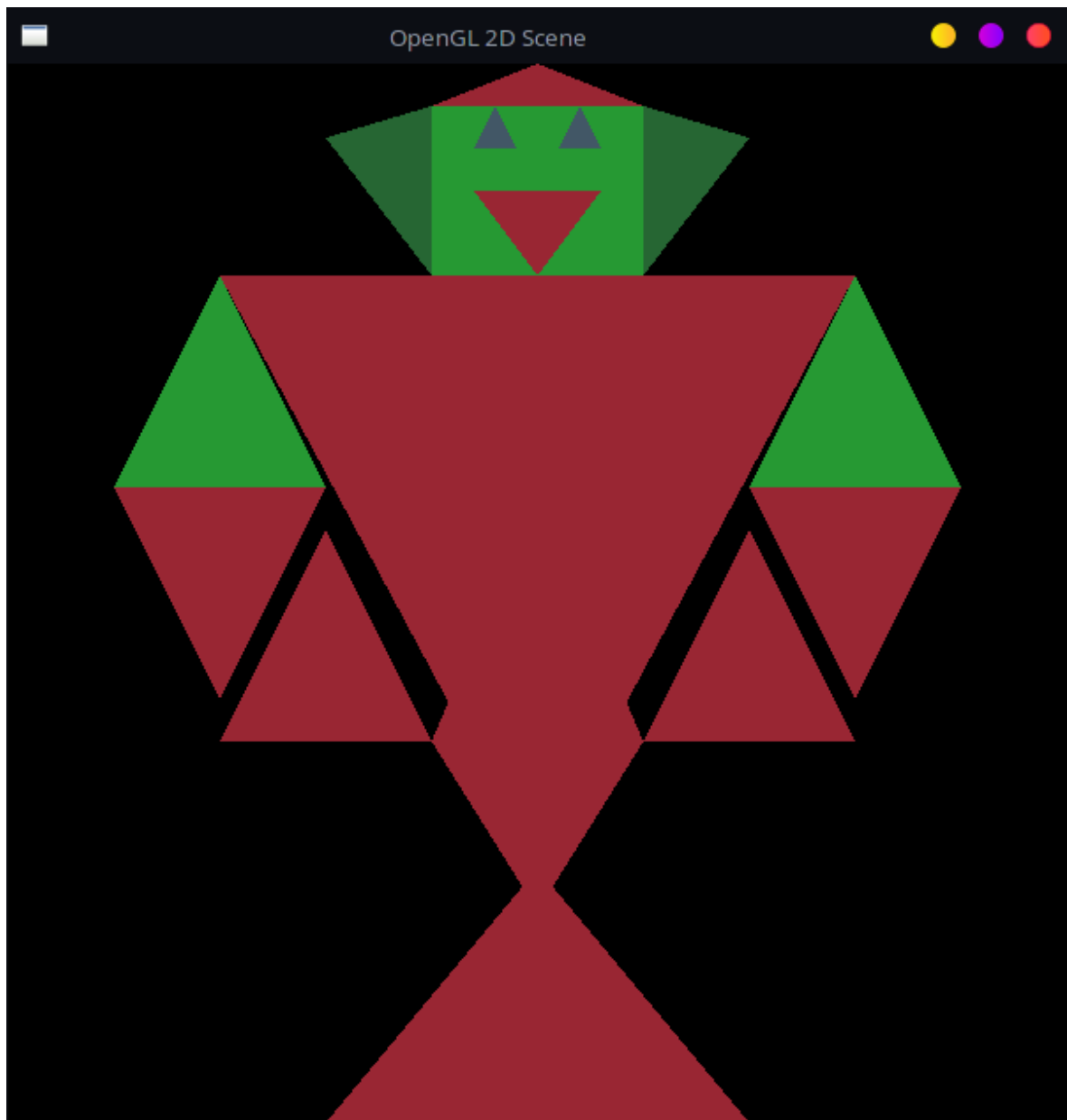
Με την εντολή `glDrawArrays(GL_TRIANGLES, 0, 60)` χαράζονται τα τρίγωνα έτσι ώστε να παραχθεί η τελική διδιάστατη σκηνή. Το όρισμα 60 δείχνει πόσες συνολικά είναι οι κορυφές της σκηνής (σχήμα 1).



Σχήμα 1: Απλή διδιάστατη σκηνή

### 3 Χάραξη διδιάστατης σκηνής σε τριπλάσια ευκρίνεια

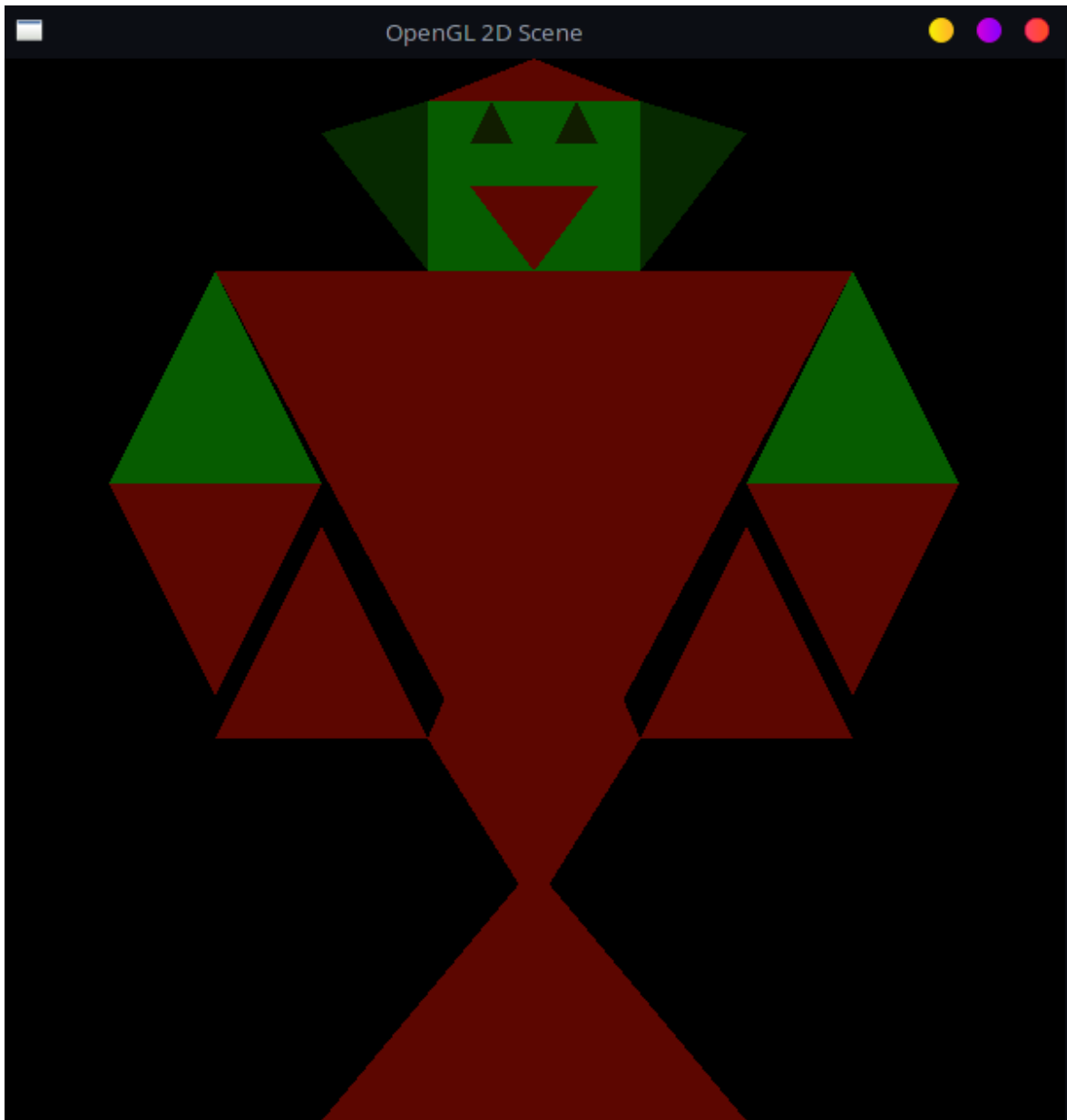
Θα πρέπει να χαραχθεί η ίδια διδιάστατη σκηνή σε τριπλάσια ευκρίνεια ( $s = 3$ ). Αυτό γίνεται πολύ εύκολα, απλά αντικαθιστώντας τις τιμές του μήκους και του πλάτους του παραθύρου με τις τριπλάσιες. Μπορούμε είτε να αλλάξουμε τις τιμές στις τριπλάσιες κατά της αρχικοποίηση των μεταβλητών αυτών. Εναλλακτικά μπορούμε να πολλαπλασιάσουμε τις ήδη υπάρχουσες τιμές ύψους και πλάτους με το τρία (σχήμα 2).



Σχήμα 2: Διδιάστατη σκηνή σε τριπλάσια ευκρίνεια

## 4 Μεταδιήθηση με γκαουσιανό ηθμό συνέλιξης

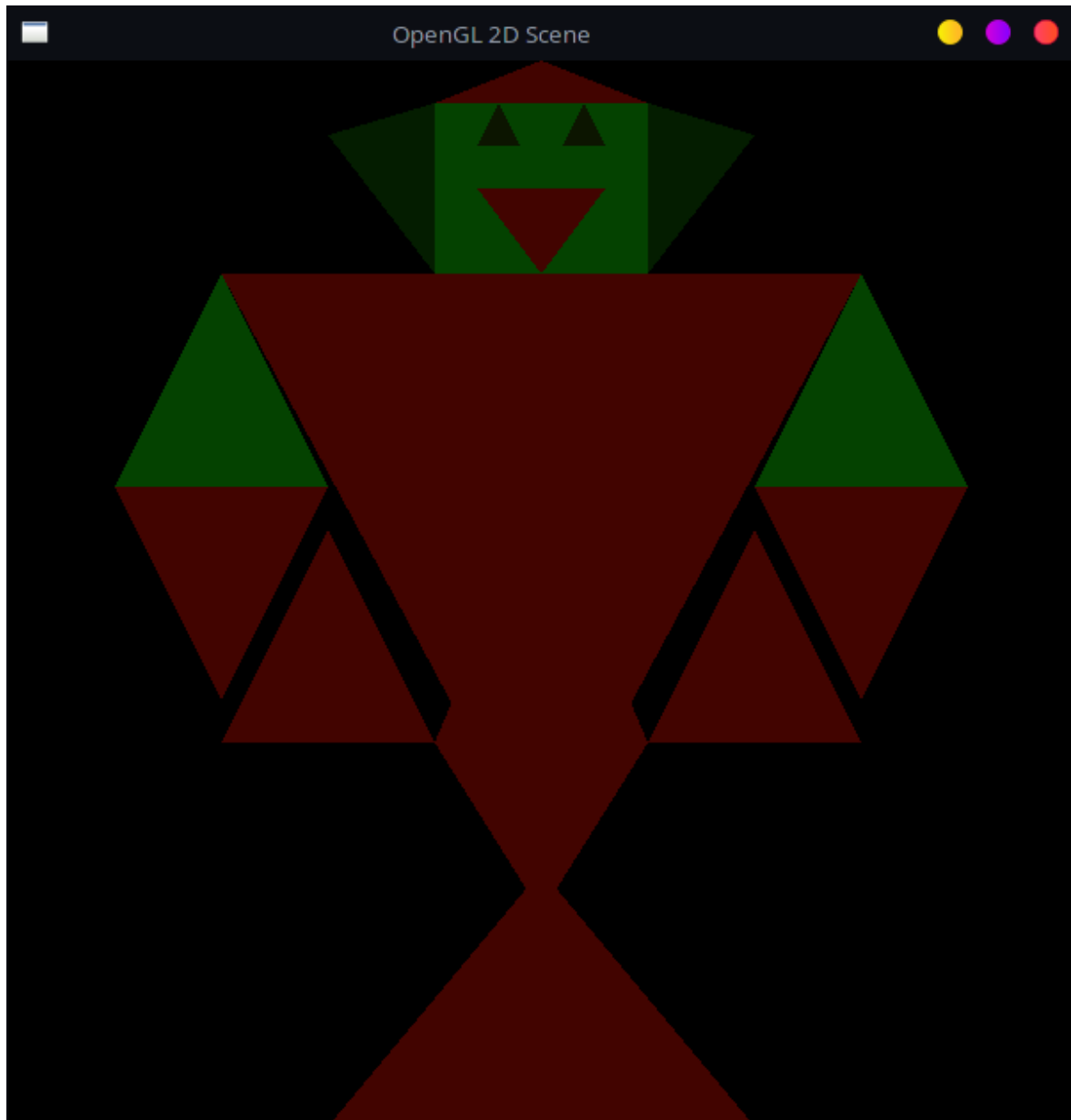
Κατά τη διαδικασία της συνέλιξης ο ηθμός διατρέχει την εικόνα και κάθε φορά υπολογίζεται το άθροισμα των γινομένων των στοιχείων του ηθμού με τα αντίστοιχα εικονοστοιχεία (pixels) της εικόνας. Στην εικόνα εξόδου το αποτέλεσμα της συνέλιξης αποδίδεται στο κεντρικό pixel του παραθύρου. Για να επιτευχθεί αυτή η διαδικασία πρέπει απλά να τοποθετήσουμε σε σχόλιο την γραμμή 166 (όπου καλείται ο fragment shader με τα κανονικά χρώματα της σκηνής), και να ενεργοποιηθεί η γραμμή 167 όπου καλείται το αποτέλεσμα της διαδικασίας της συνέλιξης με τον γκαουσιανό ηθμό 3x3(σχήμα 3).



Σχήμα 3: Αποτέλεσμα μεταδιήθησης με γκαουσιανό ηθμό συνέλιξης

## 5 Μεταδιήθηση με ηθμό με όλα τα βάρη ίσα

Προκειμένου να εκτελεστεί η συνέλιξη της εικόνας με τον ηθμό ο οποίος έχει όλα τα βάρη του ίσα το μόνο που χρειάζεται να γίνει είναι να τοποθετηθεί σε σχόλιο η γραμμή 167 και να ενεργοποιηθεί η γραμμή 168 όπου καλείται το αποτέλεσμα της συνέλιξης με τον 3x3 ηθμό με όλα τα βάρη ίσα (σχήμα 4).



Σχήμα 4: Αποτέλεσμα μεταδιήθησης με ηθμό με όλα τα βάρη ίσα



## 6 Συμπεράσματα

Αρχικά η χάραξη της διδιάστατης σκηνής γίνεται σε παράθυρο διαστάσεων 200x200. Ακολούθως οι διαστάσεις του παραθύρου της σκηνής και κατ' επέκταση η ευκρίνειά της αυξάνονται καθώς πολλαπλασιάζονται το πλάτος και το ύψος του παραθύρου με το 3. Η συνέλιξη της σκηνής με τον γκαουσιανό ηθμό οδήγησε σε μεταβολές των χρωμάτων της εικόνας. Υπάρχει επίσης μείωση στην ένταση της φωτεινότητας. Η συνέλιξη της σκηνής με τον ηθμό που έχει όλα τα βάρη ίσα, έχει ως αποτέλεσμα παρόμοιες χρωματικές αλλαγές με αυτές τις πρώτης συνέλιξης. Το περιεχόμενο της εικόνας είναι όμως λιγότερο διακριτό καθώς έχει μειωθεί η ένταση της φωτεινότητας ακόμη περισσότερο. Και στις δύο περιπτώσεις συνέλιξης, η σκηνή είναι λιγότερο ξεκάθαρη, τα χρώματα αλλάζουν χωρίς όμως να υπάρχουν μεγάλες διακυμάνσεις.

## 7 Επίλογος

Εν κατακλείδι, η διδιάστατη σκηνή υλοποιήθηκε σε γλώσσα C με χρήση της βοηθητικής βιβλιοθήκης glfw γραφικών OpenGL. Ακολούθησε μια ανάλυση του πηγαίου κώδικα που υλοποιήθηκε για την χάραξη της διδιάστατης σκηνής, σε μια αρχική και στην συνέχεια σε τριπλάσια ευκρίνεια. Έπειτα εκτελέστηκε μεταδιήθηση με γκαουσιανό ηθμό συνέλιξης 3x3, αλλά και με ηθμό συνέλιξης 3x3 ο οποίος είχε όλα τα βάρη ίσα. Με βάση τα κριτήρια της εργασίας αλλά και τα αποτελέσματα κάθε διαδικασίας παρουσιάστηκαν τα τελικά συμπεράσματά μας.

## Γλωσσάρι

**glsl** OpenGL Shading Language. 3

**OpenGL** Open Graphics language. 2

**RGB** Χρωματικός χώρος (Κόκκινο, Πράσινο, Μπλε). 3

**ηθμός** Πίνακας φίλτρο. 6

**συνέλιξη** Μαθηματική πράξη μεταξύ δύο συναρτήσεων. 7

## Αναφορές

[1] Hearn Baker Carithers. *Computer Graphics with Open GL*. PEARSON, 4th edition, 2014.

- [2] Joey de Vries. learn opengl. <https://learnopengl.com/>.
- [3] glfw development team. glfw an opengl library. <https://www.glfw.org/>.
- [4] Θάνος Τσακίρης Δημήτριος Τζοβάρας Κωνσταντίνος Μουστάκας, Ιωάννης Παλιόκας.  
*Γραφικά και εικονική πραγματικότητα*. ΣΕΑΒ, 2015.