

Art Gallery  
Γραφικά και Εικονική Πραγματικότητα

ΚΟΣΜΑΣ ΑΡΧΟΝΤΗΣ, 1084020

Πανεπιστήμιο Πατρών

25 Σεπτεμβρίου 2024

# Περιεχόμενα

<b>1 Εισαγωγή</b>	<b>2</b>
<b>2 Περιγραφή του Προβλήματος</b>	<b>3</b>
<b>3 Μεθοδολογία</b>	<b>5</b>
3.1 Μέρος A . . . . .	5
3.1.1 Ερώτημα 1 . . . . .	5
3.1.2 Ερώτημα 2 . . . . .	9
3.1.3 Ερώτημα 3 . . . . .	17
3.2 Μέρος B . . . . .	20
3.2.1 Ερώτημα 4 . . . . .	20
3.2.2 Ερώτημα 5 . . . . .	36
3.2.3 Ερώτημα 6 . . . . .	40
3.2.4 Ερώτημα Extra - Transitions . . . . .	45
<b>4 Θεωρία/Θεωρητικό Υπόβαθρο</b>	<b>51</b>
4.1 Normal Mapping . . . . .	51
4.2 Parallax Mapping . . . . .	51
4.3 Weighted Voronoi Stippling . . . . .	52
4.4 Sobel filters . . . . .	53
<b>5 Αποτελέσματα &amp; Συζήτηση</b>	<b>54</b>
5.1 Αποδόσεις . . . . .	54
5.2 Αδυναμίες - Πιθανές Βελτιώσεις . . . . .	54
<b>6 Συμπεράσματα</b>	<b>55</b>
<b>7 Οδηγίες Χρήσης Προγράμματος</b>	<b>56</b>
7.1 Για τη χρήση εντός του προγράμματος: . . . . .	56
<b>8 Βιβλιογραφία</b>	<b>58</b>

# 1 Εισαγωγή

Σκοπός αυτής της αναφοράς είναι η περιγραφή της διαδικασίας εκπόνησης και αποτελεσμάτων της απαλλακτικής εργασίας για το μάθημα Γραφικά και Εικονική Πραγματικότητα του τμήματος Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών.

Η θεματολογία της εργασίας περιλαμβάνει διαχείριση πολλαπλών μοντέλων (μοντέλα-δωμάτια), πολλαπλών framebuffers και τα αντίστοιχα attached textures τους, post-processing effects για το κάθενα, bump mapping με παραγωγή και χρήση normal maps και δημιουργία ανάγλυφων αναπαραστάσεων δισδιάστατων επιφανειών.

Το πρόγραμμα που έχει αναπτυχθεί είναι γραμμένο σε OpenGL/C++, GLSL για προγραμματισμό στους shaders, κάνοντας χρήση των βιβλιοθηκών GLEW, GLFW και GLM για version, context handling και μαθηματικές πράξεις.

## 2 Περιγραφή του Προβλήματος

Η κεντρική ιδέα της εργασίας είναι η δημιουργία μιας πινακοθήκης/γκαλερί, όπου ο χρήστης μπορεί να “μπει μέσα” στους πίνακες και να τηλεμεταφερθεί στη σκηνή-δωμάτιο που απεικονίζουν. Δημιουργούμε εικονικές πύλες (portals) οι οποίες δημιουργούν ένα “παράθυρο” στο κεντρικό κυλινδρικό δωμάτιο εντός των άλλων δωματίων και θα μπορεί ο χρήστης να βλέπει μέσα στο καθένα. Κάθε πίνακας θα σχετίζεται με ένα ιδιαίτερο εφέ/στύλο το οποίο θα εφαρμόζεται πάνω του και μέσα στο δωμάτιο, και θα μπορεί ο χρήστης να παρατηρήσει.

Το αργότερο μέρος της εργασίας σχετίζεται με απεικόνιση αναγλύφου, τις λεγόμενες τεχνικές bump mapping & normal mapping για πιο ρεαλιστικό και ενδιαφέρον αποτέλεσμα και ανάδραση με πηγές φωτισμού απλών δισδιάστατων textures, καθώς και υλική ανύψωση αυτών, σύμφωνα με κάποιον χάρτη ύψους.

Παρατίθεται αναλυτικότερα ολόκληρη η λίστα με τα ζητούμενα της εργασίας:

## 6. Art gallery

### Part A:

1. Create a cylindrical room with 5 paintings equally spaced around the wall. Each painting will be associated with a different room/scene. The user can enter the painting and get into the respective room.
2. Create the rooms, add objects, lighting and shadows.
3. Each room will have a unique art style. The art style will be applied on the resulting frame, before showing it to the user. Create 5 separate functions/shaders that will handle the frame buffers of each of the different rooms.

### Part B:

4. a. Room 1: Implement Floyd-Steinberg dithering, to quantize the colors in the frame.
  - b. Room 2: Create a brush stroke effect.
  - c. Room 3: Instead of solid color, use small circles of various sizes (Pointilism)
  - d. Room 4: Implement a grayscale, comic book effect.
  - e. Room 5: Choose a different style according to your preferences.
5. Generate a bump map based on the normals of the scenes and use them for bump renderings of the paintings
  6. Use a depth map to raise the mesh of the paintings and generate an anaglyph representation

# **3 Μεθοδολογία**

Σε αυτή την ενότητα θα περιγράψουμε τη διαδικασία επίλυσης των ερωτημάτων, τον τρόπο σκέψης και θα περιγράψουμε επίσης την ανάπτυξη του προγράμματος χρονολογικά. Θα αναφερθούν περιληπτικά αρχικές σκέψεις και αποτυχημένες προσπάθειες, καθώς και πόση δουλειά και πόσα ερωτήματα όλοκληρώθηκαν εν τέλει.

## **3.1 Μέρος Α**

### **3.1.1 Ερώτημα 1**

Έχοντας έτοιμη τη θεμελίωση της εκτέλεσης από τον κώδικα των εργαστηρίων του μαθήματος, πρώτον πρέπει να βρούμε τα μοντέλα του κεντρικού δωματίου και των πινάκων.

Κάνουμε ευρεία αναζήτηση με τις λέξεις κλειδιά στην ιστοσελίδα τρισδιάστατων μοντέλων <https://sketchfab.com>, και αποφασίζουμε στο παρακάτω μοντέλο για τη κορνίζα των πινάκων:



Σχήμα 3.1: Το μοντέλο των πινάκων του χρήστη boo123, picture frame.

Και για το κεντρικό δωμάτιο γκαλερί, παρότι δεν είναι κυλινδρικό:

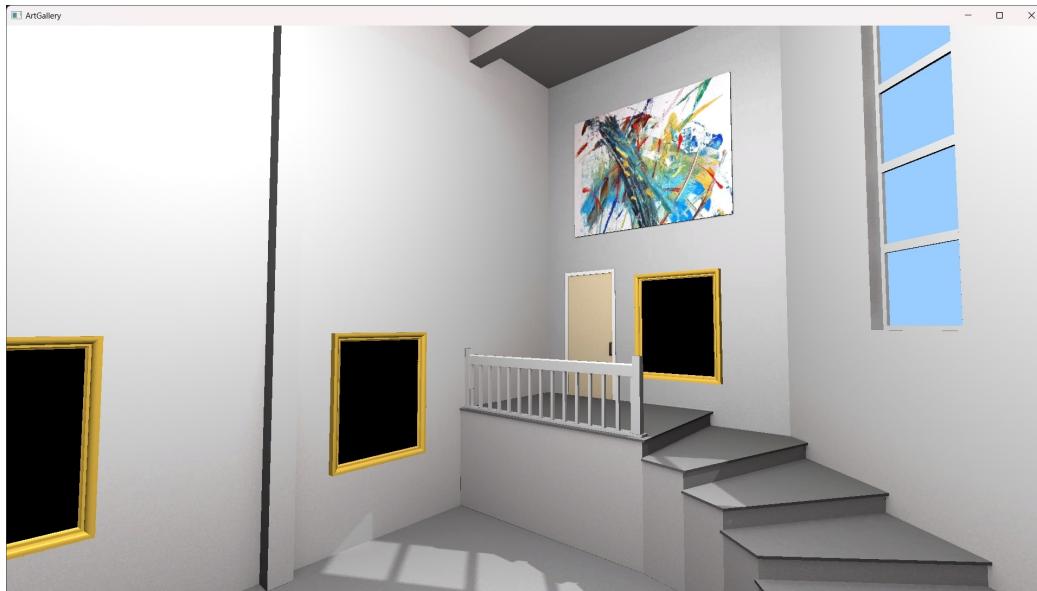


Σχήμα 3.2: Το μοντέλο του κεντρικού δωματίου του χρήστη jimbogies, Studio Room.

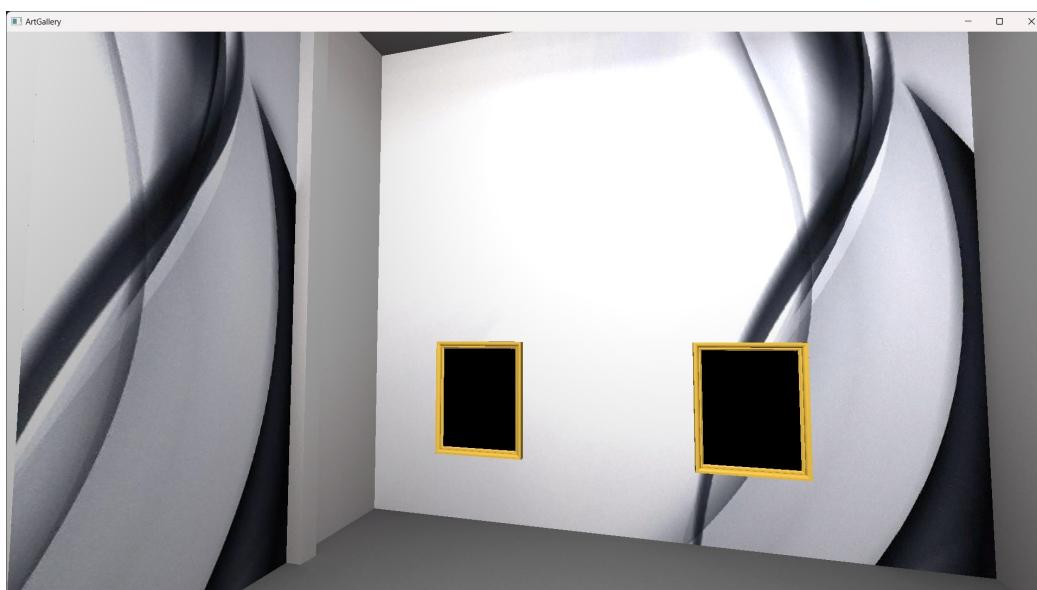
Το οποίο συνοδεύεται από τα textures: diffuse, emission και roughness.

Πάνω από τους πίνακες επίσης τοποθετούμε ένα απλό ορθογώνιο, στο οποίο ύστερα θα απεικονίσουμε το texture του portal.

Μέσα από τη C++ φτιάχνουμε τις τοποθεσίες και προσανατολισμούς των μοντέλων στον 3Δ χώρο, και κάνουμε render τη σκηνή μας.



Σχήμα 3.3: Τοποθέτηση των 3 πλαισίων πινάκων στο Studio Room.

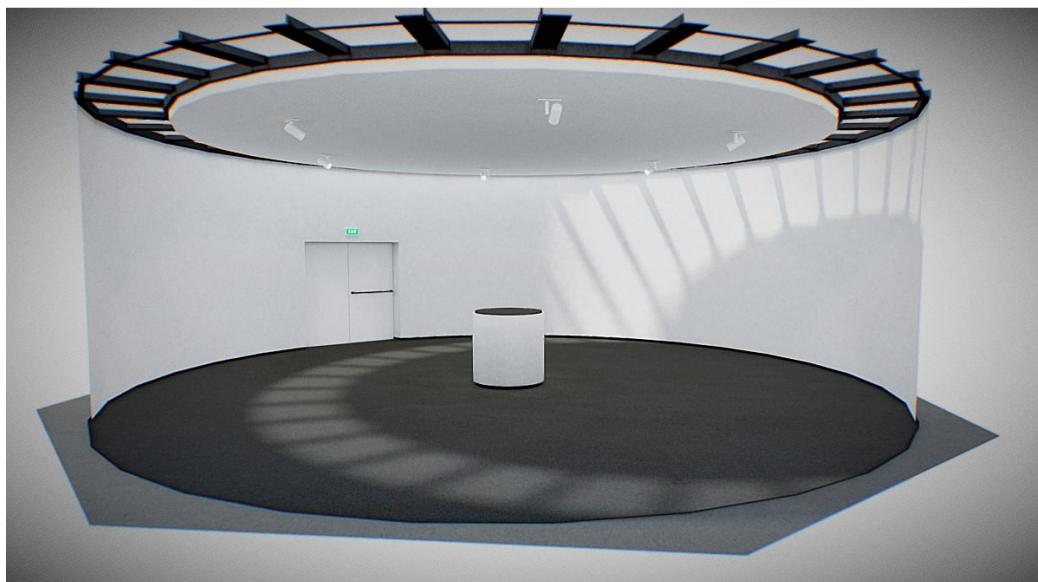


Σχήμα 3.4: Τοποθέτηση των υπόλοιπων 2 στο Studio Room.

Μερικά εναλλακτικά δωμάτια που εξετάσαμε για το κεντρικό δωμάτιο, είναι τα παρακάτω:



Σχήμα 3.5: Πιθανό μοντέλο για κεντρικό δωμάτιο του χρήστη Lora, lighthouse.



Σχήμα 3.6: Πιθανό μοντέλο για κεντρικό δωμάτιο του χρήστη Maxim Mavrichev, VR Round Art Gallery.

Αλλά τελικά δεν συμπεριλήφθηκαν στο πρόγραμμά μας.

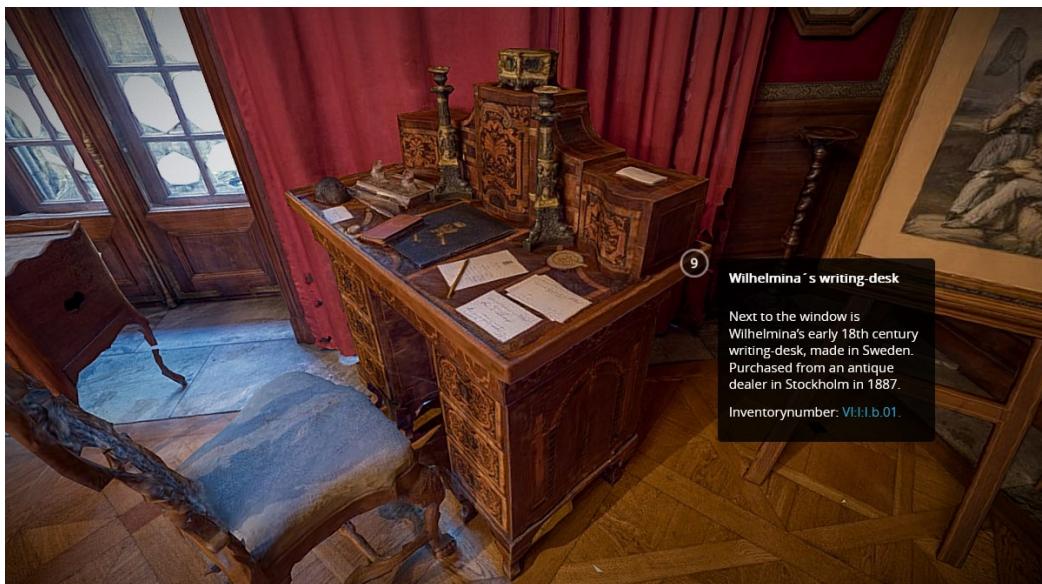
### 3.1.2 Ερώτημα 2

Έχοντας έτοιμο το κεντρικό δωμάτιο, αναζητούμε πλέον τα 5 διαφορετικά μοντέλα-δωμάτιο που αντιστοιχούν σε κάθε έναν από τους πίνακες. Ύστερα από εκτενή αναζήτηση, καταλήξαμε στα παρακάτω 5 μοντέλα.

Αρχικά, τα πρώτα 2 δωμάτια τα λαμβάνουμε από το μουσείο Hallwyl της Στοκχόλμης. Και τα δύο αποτελούν αναπαραστάσεις πραγματικών δωματείων του χώρου, ύστερα από σάρωση και ψηφιοποίηση. Το πρώτο είναι το Πρωινό Δωμάτιο (The Morning Room):



Σχήμα 3.7: Το Morning Room μοντέλο, προσφορά του Μουσείου Hallwyl.



Σχήμα 3.8: Λεπτομέρεια του μοντέλου The Morning Room.

Και το δεύτερο πρόκειται για το Μεγάλο Δωμάτιο Ζωγραφικής (Great Drawing Room):

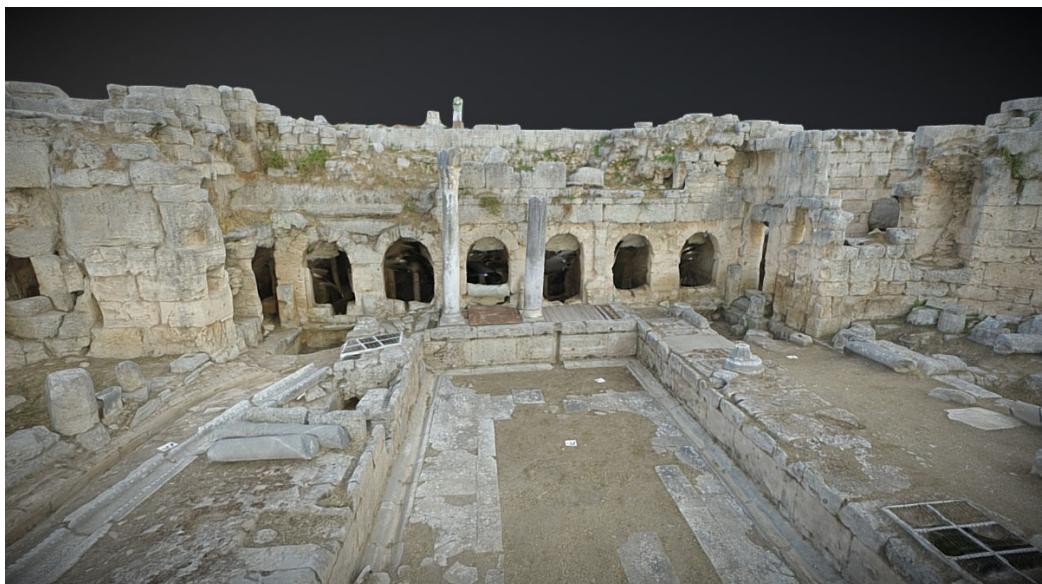


Σχήμα 3.9: To Great Drawing Room μοντέλο, προσφορά του Μουσείου Hallwyl.

Το τρίτο δωμάτιο που θα μας απασχολήσει είναι ένα 3Δ μοντέλο της κρήνης της Πειρήνης, αρχαιολογικός χώρος της Κορίνθου και σημαντική πηγή της αρχαιότητας.



Σχήμα 3.10: Φωτογραφία της Κρήνης της Πειρήνης.



Σχήμα 3.11: Το 3Δ μοντέλο Peirene Fountain, Corinth - Greece, του χρήστη Miguel Bandera.

Το τέταρτο δωμάτιο είναι ένα μοντέλο Γκρεμισμένου Δωματίου, Destroyed

Room:

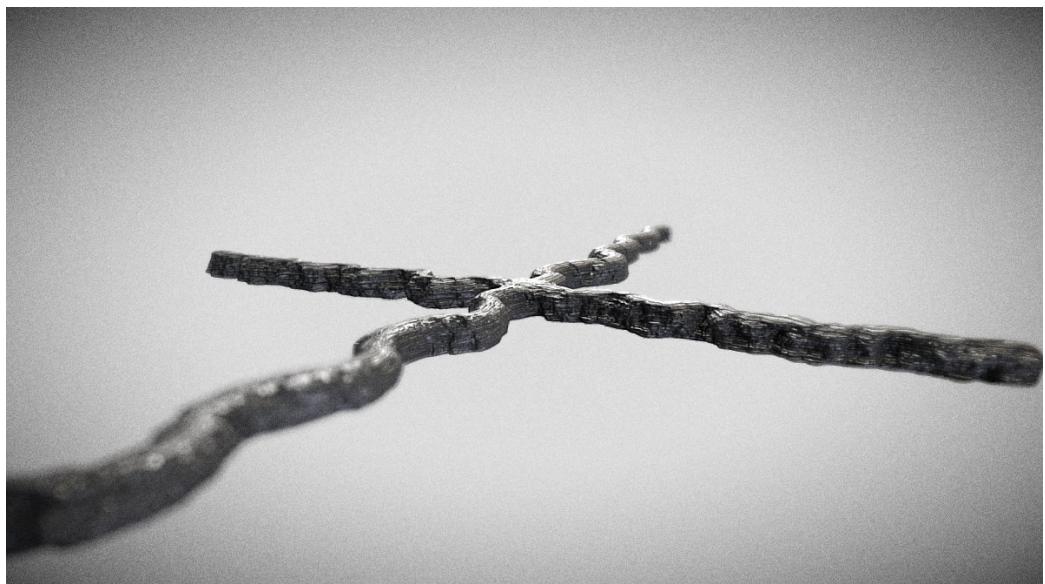


Σχήμα 3.12: Το 3Δ μοντέλο Destroyed Rooms, του χρήστη jimbogies.

Το πέμπτο και τελευταίο δωμάτιο είναι η Μεταλλική Σπηλιά, Metallic Cave, μοντέλο του χρήστη natanbolek στο Sketchfab:



Σχήμα 3.13: Το 3Δ μοντέλο Metallic Cave, του χρήστη natanbolek.

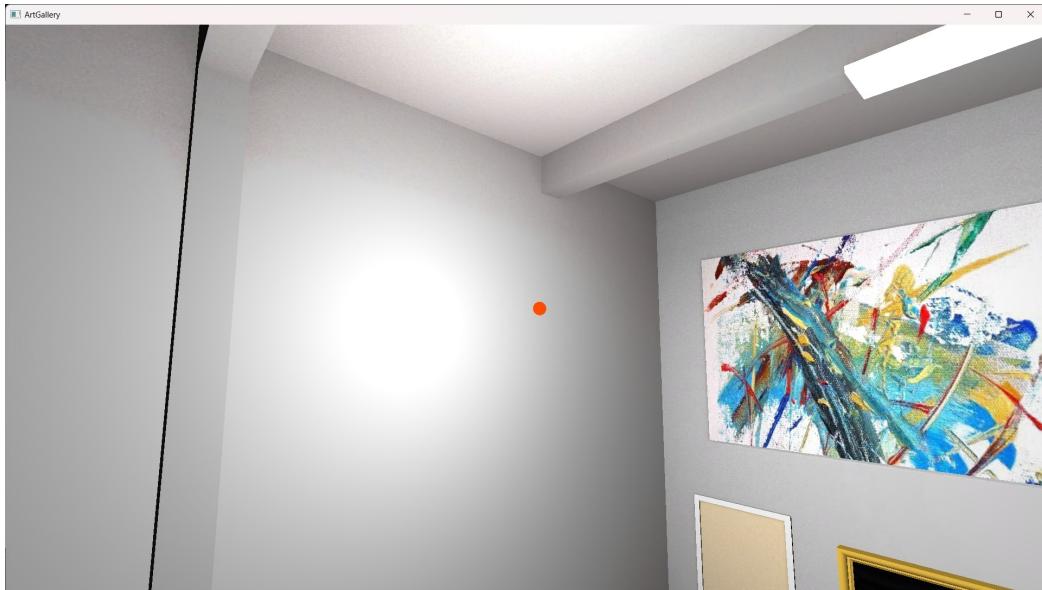


Σχήμα 3.14: Το ιδιαίτερο layout του μοντέλου.

Το συγκεκριμένο μοντέλο έρχεται μαζί με τα textures: ambient occlusion, roughness και normal.

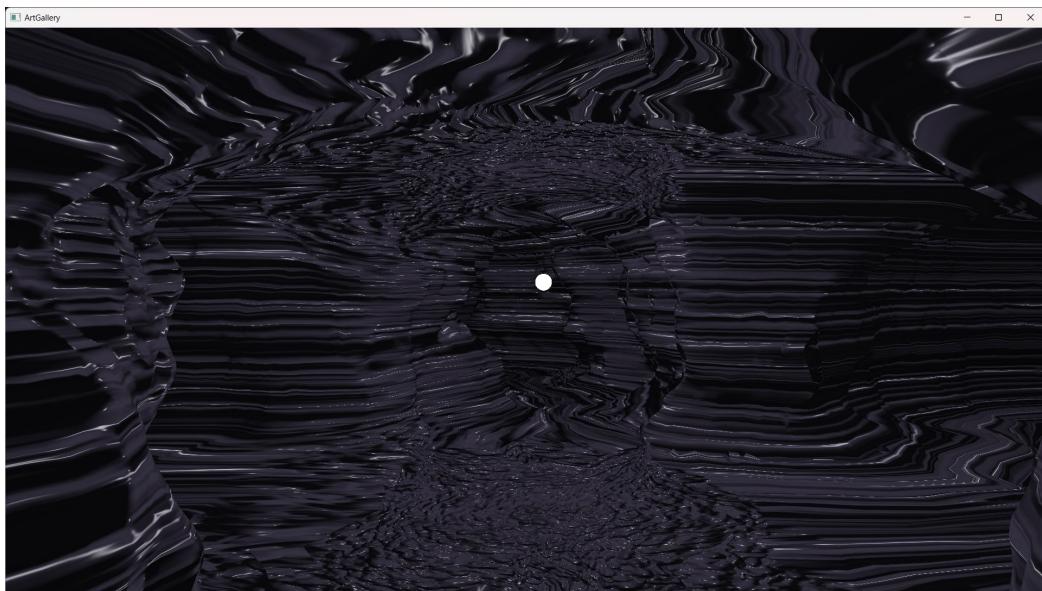
Το Studio Room και Metallic Cave απαιτούν εφαρμογή φωτισμού ώστε να μπορούν να γίνουν render σωστά, επομένως εφαρμόζουμε φωτισμό Phong για point lights.

Τοποθετούμε ένα κεντρικό φώς στο Studio Room, το οποίο διαγράφει μια κυκλική τροχιά πάνω από τους πίνακες για λόγους που θα φανούν σε επόμενο ερώτημα.

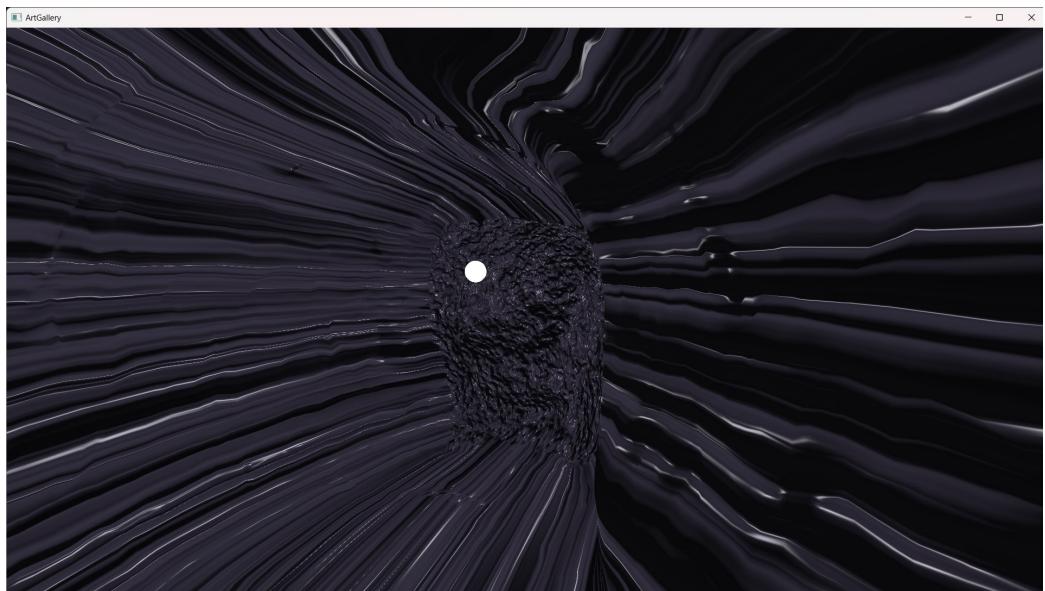


Σχήμα 3.15: Το περιστρεφόμενο φως του κεντρικού δωματίου περιβαλλόμενο από μια πορτοκαλί σφαίρα για οπτικοποίηση.

Για τη Μεταλλική Σπηλιά, τοποθετούμε μερικά παραπάνω φώτα στις άκρες, και για περαιτέρω δυναμικό εφέ τα αφήνουμε να δονίζονται ελαφρώς. Πρέπει να εφαρμόσουμε και normal mapping [4.1] για το Normal Texture που μας δίνεται, όπως αναφέραμε πιο πριν, και αυτό θα γίνει στους shaders.



Σχήμα 3.16: Το κεντρικό φως στο δωμάτιο της Μεταλλικής Σπηλιάς.



Σχήμα 3.17: Πλαϊνό φως στο μοντέλο Metallic Cave.

Παρουσιάζουμε τελικά τα δωμάτια, όπως τα κάνουμε render στη σκηνή μας:



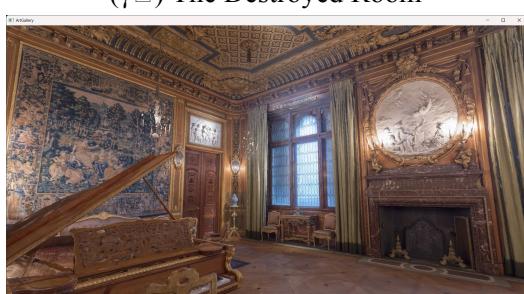
( $\alpha \square$ ) The Morning Room



( $\beta \square$ ) Peirene Fountain



( $\gamma \square$ ) The Destroyed Room



( $\delta \square$ ) The Great Drawing Room



( $\varepsilon \square$ ) Metallic Cave

$\Sigma\chi\eta\mu\alpha$  3.18: All of the rooms rendered

### 3.1.3 Ερώτημα 3

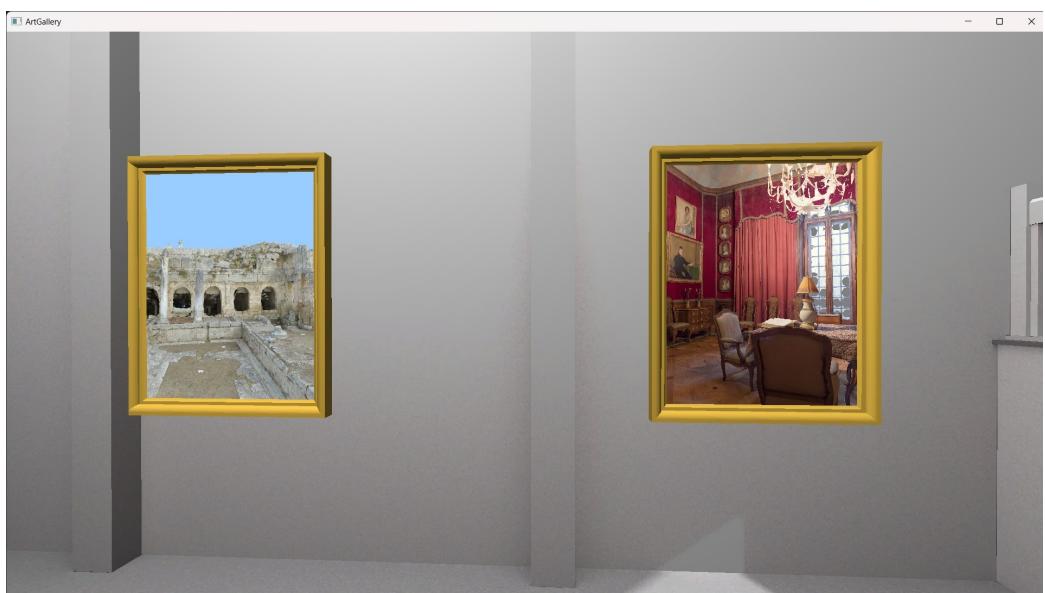
Στο τρίτο ερώτημα μας ζήτείται να κατασκευάσουμε ξεχωριστές συναρτήσεις/shaders που θα χειρίζονται τους 5 ξεχωριστούς framebuffers για τα post-processing effects στα δωμάτια, καθώς και να παρουσιαστούν στους πίνακες που βλέπει ο χρήστης. Για λόγους καλύτερης σύγκρισης μεταξύ των εφέ και των δωματίων, θα επιτρέψουμε να χρησιμοποιείται οποιοδήποτε εφέ σε οποιοδήποτε δωμάτιο - που σημαίνει ότι πρέπει να βρίσκονται στον ίδιο shader.

Επομένως, κατασκευάζουμε 5 framebuffers, ο καθένας με δικό του attached texture και depth component texture. Αυτό το texture θα αντιστοιχεί στην εικόνα της default κάμερας που χειρίζεται ο χρήστης, στο οποίο θα εφαρμοστούν αργότερα τα post-processing effects, και θα παρουσιάζεται εκ νέου σε ένα screen-wide quad.

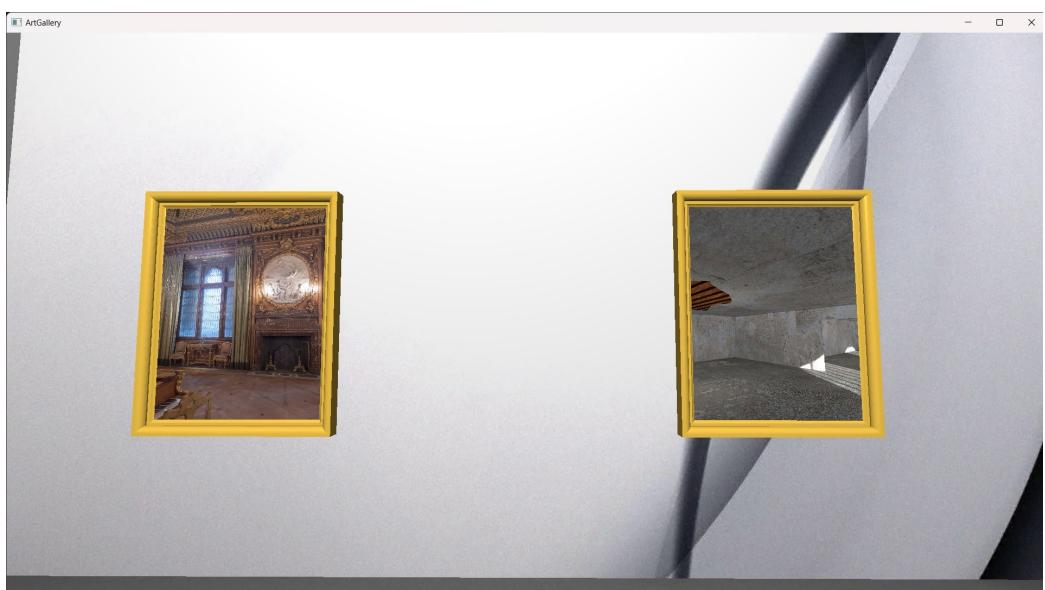
Όμως, χρειαζόμαστε επιπλέον 5 framebuffers, 5 attached textures και depth components για να κατασκευάσουμε τα portals στο εκάστοτε δωμάτιο. Το αντίστοιχο εκείνο texture θα αναπαραστά τη "βλέψη" μιας εικονικής κάμερας που θα βρίσκεται μέσα στο δωμάτιο και θα απεικονίζεται πάνω στο ορθογώνιο που τοποθετήσαμε νωρίτερα πάνω στο painting frame της σκηνής.

Άρα, αν βρισκόμαστε μέσα στο Studio Room, πρέπει να κάνουμε ένα render pass για το ίδιο το δωμάτιο, όπως στο Ερώτημα 1 [3.1.1], συν 5 για κάθε πίνακα-δωμάτιο. Τέλος πρέπει επίσης να ζωγραφιστούν οι εικόνες αυτές των 5 δωματίων πάλι στο κεντρικό δωμάτιο, όπου όλα αυτά γίνονται rendered στον framebuffer που αντιστοιχεί στο Studio Room. Στον default framebuffer απλώς κάνουμε render το quad έχοντας ως texture αυτού την προαναφερθείσα σκηνή.

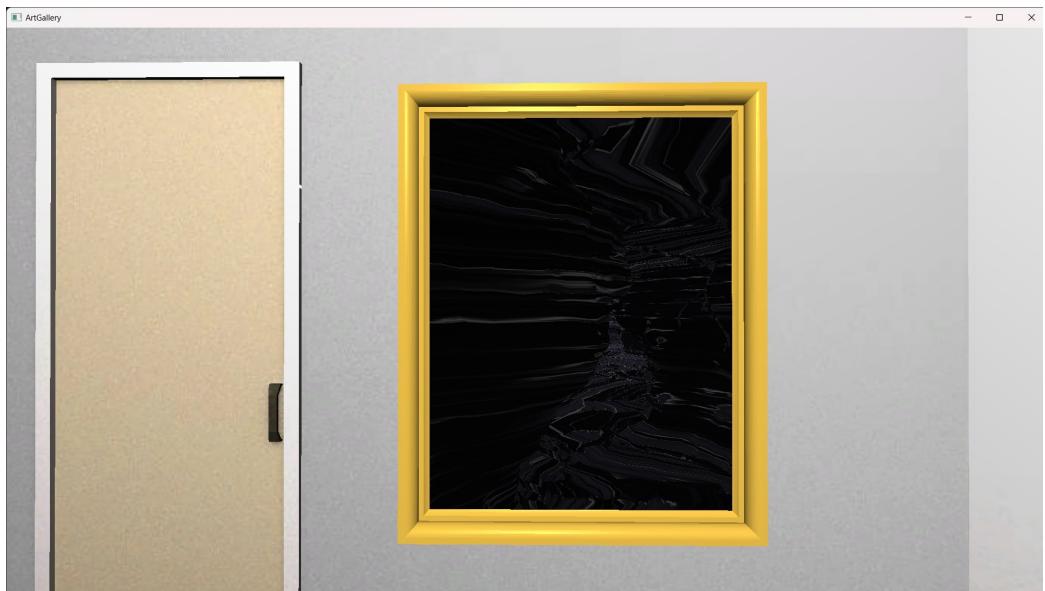
Όταν τηλεμεταφέρομαστε στο εσωτερικό ενός δωματίου, το rendering γίνεται στο δικό του framebuffer και στον default απλώς γίνεται render το quad όπως και πριν. Λαμβάνουμε τα παρακάτω αποτελέσματα:



Σχήμα 3.19: Τα portals των δωματίων Morning Room και Peirene Fountain.

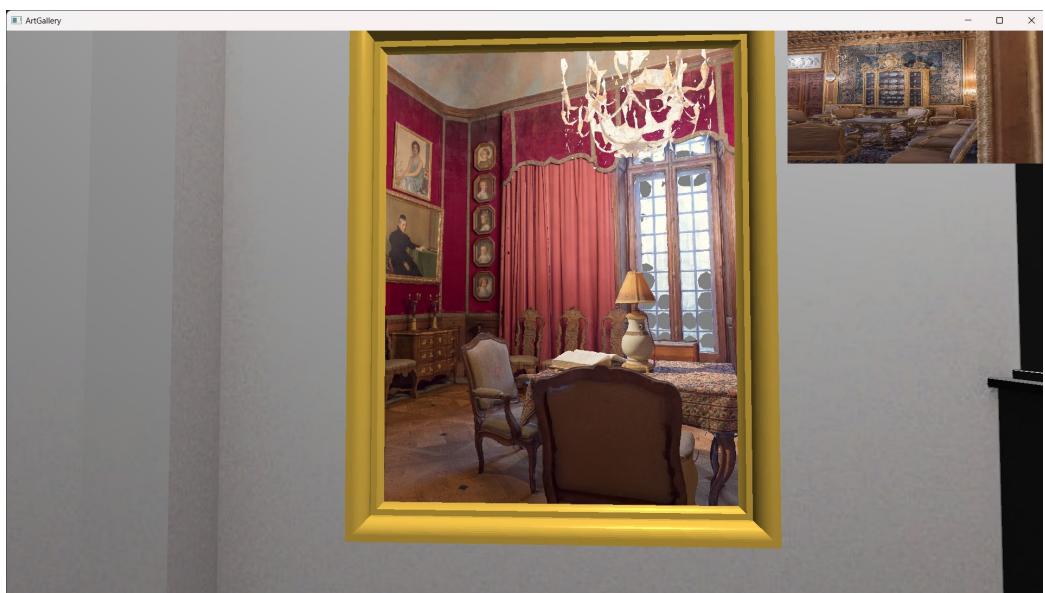


Σχήμα 3.20: Τα portals των δωματίων Destroyed Room και Great Drawing Room.



Σχήμα 3.21: Το portal του δωματίου Metallic Cave.

Και για οπτικοποίηση των διαφορετικών framebuffers, κάνουμε render ένα miniquad πάνω αριστερά στην οθόνη, όπου φαίνεται το texture ενός portal ενώ κοιτάμε κάποιο εντελώς ξεχωριστό:



Σχήμα 3.22: 2 διαφορετικές απεικονίσης λόγω 2 διαφορετικών framebuffer.

Επίσης, επιλέγουμε οι όψεις των εικονικών κάμερων να κοινούνται σύμφω-

να με τον προσανατολισμό και την κατεύθυνση της κάμερας του χρήστη. Δε δημιουργούνται πάντα αληθιοφανή αποτελέσματα, αλλά μπορεί να διατηρηθεί η ψευδαίσθηση υπαρκτού “παραθύρου” σε άλλο δωμάτιο υπό κάποιες γωνίες.

Έχοντας ολοκληρώσει το Α Μέρος της εργασίας, μπορούμε να προχωρήσουμε στην επέκταση του προγράμματος με την εφαρμογή των 5 διαφορετικών στυλ, και στο Β Μέρος.

## 3.2 Μέρος Β

### 3.2.1 Ερώτημα 4

Στο ερώτημα αυτό ζητείται να υλοποιήσουμε τα 5 διαφορετικά στυλ των πινάκων-δωμάτιων.

#### a. Floyd-Steinberg Dithering

Για το πρώτο δωμάτιο ζητείται να εφαρμόσουμε Floyd-Steinberg Dithering για κβάντωση των χρωμάτων στον πίνακα.

Το dithering ως τεχνική χρησιμοποιείται για εξοικονόμηση αποθηκευτικού χώρου, μειώνοντας τα υπαρκτά χρώματα σε μια είκονα. Πλέον, προτιμάται για την αισθητική που αποδίδει στις εικόνες. Το dithering θα εφαρμοστεί στο texture που είναι attached στον binded framebuffer εκείνη τη στιγμή.

Τα συγκεκριμένα βήματα χοντρικά του Floyd-Steinberg Dithering είναι τα εξής:

- Αρχικοποιούμε buffers σφαλμάτων για κάθε κανάλι από τα RGB (εφόσον είναι για έγχρωμο dithering).
- Διατρέχουμε κάθε πίξελ της εικόνας.
- Λαμβάνουμε το χρώμα του τωρινού πίξελ
- Πραγματοποιούμε κβάντωση στη κοντινότερη επιτρεπτή τιμή, ανάλογα τα bit κβάντωσης
- Υπολογίζουμε το σφάλμα κβάντωσης
- Διανέμουμε το σφάλμα στα γειτονικά πίξελ. Συγκεκριμένα 7/16 του σφάλματος στο δεξί, 3/16 στο κάτω αριστερά, 5/16 στο κάτω, 1/16 στο κάτω δεξιά.
- Επαναλαμβάνουμε για όλα τα πίξελ

Παρατηρούμε ότι ο αλγόριθμος λειτουργεί σειριακά, μεταφέροντας το σφάλμα στα υπόλοιπα πιξέλ από πάνω αριστερά έως κάτω δεξιά. Εφαρμόζοντας το εφέ αυτό λοιπόν στη σκηνή του Morning Room, λαμβάνουμε τη παρακάτω εικόνα:

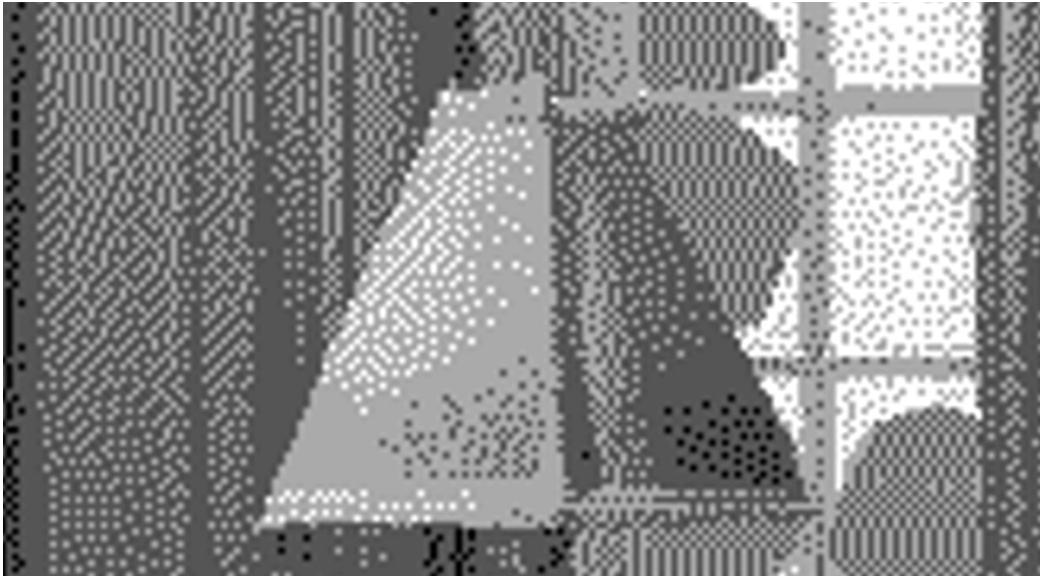


Σχήμα 3.23: Εφαρμογή του colored Floyd-Steinberg Dithering στην εικόνα του Morning Room.



Σχήμα 3.24: Εφαρμογή του grayscale Floyd-Steinberg Dithering στην εικόνα του Morning Room.

To effect είναι πετυχημένο λόγω του γεγονότος ότι δεν είναι εύκολα παρατηρήσιμο για μεγάλη ανάλυση. Αν κοιτάξουμε προσεκτικά όμως, μπορούμε να δούμε συγκεκριμένα την κβάντωση, ή και τη συσσώρευση σφάλματος σε συγκεκριμένα πίξελ. Το συγκεκριμένο grayscale dithering έχει πραγματοποιηθεί με 2 bit, δηλαδή 4 επίπεδα κβάντωσης για κάθε πίξελ.



Σχήμα 3.25: Τα κβαντισμένα χρώματα στην grayscale dithered εικόνα. Παρατηρούμε τα 4 επίπεδα μαύρο, σκούρο γκρι, ανοιχτό γκρι, λευκό.

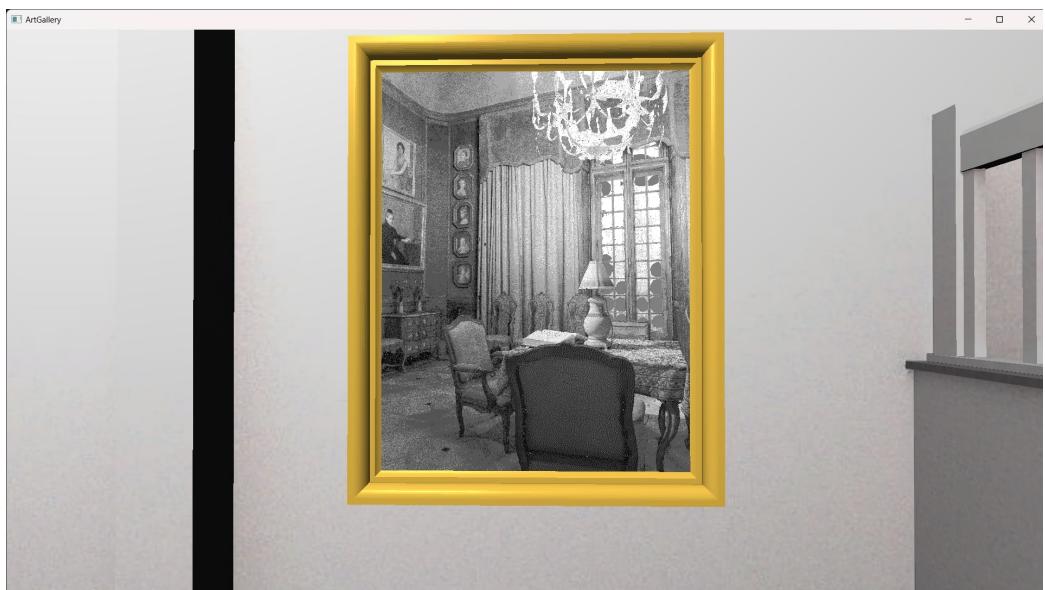
Όπως παρατηρήσαμε προηγουμένως, ο αλγόριθμος του Floyd-Steinberg Dithering είναι απολύτως σειριακός, που σημαίνει ότι δεν μπορούμε να τον πραγματοποιήσουμε στην κάρτα γραφικών, και πρέπει να λάβει χώρα στην CPU. Αυτό έχει ως αποτέλεσμα σοβαρή πτώση στα frames per second (στο υπολογιστικό σύστημα στο οποίο αναπτύχθηκε αυτή η εργασία) του προγράμματος. Ενδεικτικά, ύστερα από πολλες βελτιστοποιήσεις στο επίπεδο κώδικα για ευνόηση του τρόπου λειτουργίας του hardware (εκμετάλευση τοπικότητας, αφαίρεση αργών πράξεων, αποθήκευση αποτελεσμάτων κλπ.), το colored dithering απαιτεί περίπου 0.225 δευτερόλεπτα ανά καρέ, που μας δίνει 4.44 frames per second, ενώ το grayscale dithering περίπου 0.11 δευτερόλεπτα, που μας δίνει 10 frames per second.

Ως εναλλακτική στην πτώση των frames, στην κατάσταση όπου φορτώνει το πρόγραμμα, εκτελούμε FS Dithering σε όλο το texture του δωματίου, και κάνουμε render τη σκηνή με αυτό το texture έναντι του πραγματικού χρόνου dithering. Επίσης, αποθηκεύουμε το texture ως bmp αρχείο, εφόσον αναπτύξουμε την κατάλληλη συνάρτηση, ώστε να πληρώσουμε το χρονικό αυτό κόστος μία φορά.



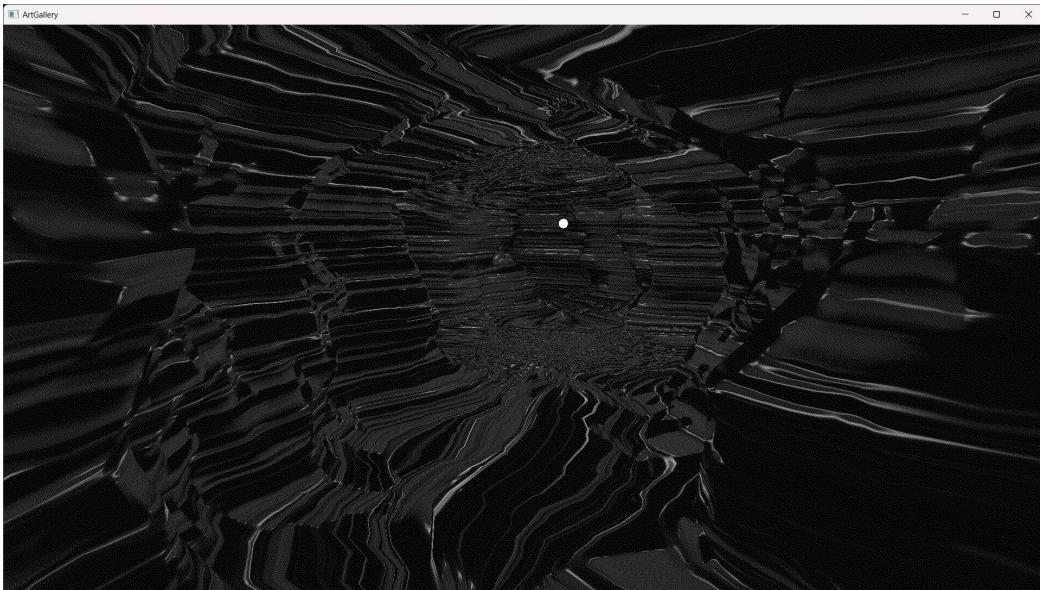
Σχήμα 3.26: Εφαρμογή του προ-dithered Morning Room texture.

Επίσης, μπορούμε να εφαρμόσουμε το dithering και στο texture που δημιουργούμε για τα portals.



Σχήμα 3.27: To dithering μέσω από το portal.

Και, ως ένδειξη, πραγματοποιούμε το Floyd-Steinberg Dithering και σε κάποιο άλλο δωμάτιο:



Σχήμα 3.28: Εφαρμογή του Floyd-Steinberg Dithering στο δωμάτιο Metallic Cave.

### b. Pointilism

Στο δεύτερο δωμάτιο, ζητείται να εφαρμόσουμε ένα στυλ πουαντιγισμού (pointilism) κύκλων διαφόρων μεγεθών.

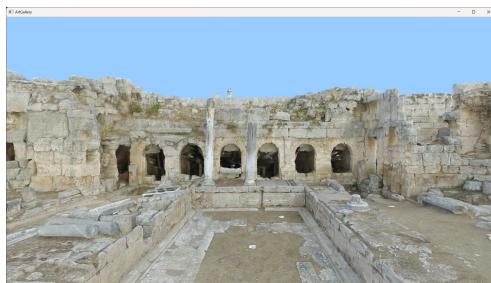
Ο πουαντιγισμός ήταν καλλιτεχνικό ρεύμα, στο οποίο πίνακες ζωγραφικής κατασκευάζονται κουκίδα κουκίδα, και υπό υψηλή ανάλυση (ή με αρκετή απόσταση) το εργό φαινόταν ομοιόμορφο. Επειδή αισθητικά θα ήταν αρκετά συγγενικό το παραγόμενο effect με αυτό του dithering, αποφασίζουμε να προσπαθήσουμε να το εφαρμόσουμε διαφορετικά, ακολουθώντας μια διαφορετική τεχνική: Αυτή του stippling.

Συγκεκριμένα, προσπαθήσαμε να εφαρμόσουμε Weighted Voronoi Stippling 4.3, η οποία χρησιμοποιεί διαγράμματα Voronoi, εύρεση του κέντρου βάρους του κελιού Voronoi και τυχαία δειγματοληψία σημείων για την ανακατασκευή εικόνας. Προσπαθήσαμε με χρήση της βιβλιοθήκης υπολογιστικής γεωμετρίας Fade2D να εφαρμόσουμε αυτή τη τεχνική, αλλά δυστυχώς το εγχείρημα απέτυχα, και νιοθετήσαμε μια ενδιάμεση προσέγγιση.

- Αρχικά κάνουμε render της σκηνής στο αντίστοιχο framebuffer του δωματίου.
- Τυχαιοκρατικά δειγματοληπτούμε από το texture κάποιο πίξελ.

- Ελέγχουμε αν το χρώμα του ικανοποιεί κάποιο όριο “σκοτεινότητας”. Αν όχι, συνεχίζουμε τη διαδικασία, αλλιώς προχωράμε στο επόμενο βήμα.
- Ως κέντρο τη θέση του πίξελ, ορίζουμε έναν κύκλο με ακτίνα ανάλογη του πόσο σκούρο είναι το αρχικό χρώμα. Όσο πιο σκούρο, τόσο μεγαλύτερη η ακτίνα.
- Όσα πίξελ πέφτουν εντός της ακτίνας, ζωγραφίζονται μαύρα. Τα υπόλοιπα άσπρα
- Επαναλαμβάνουμε σύμφωνα με έναν προαποφασισμένο αριθμό δειγματοληψίας.
- Περνάμε το modified texture στους shaders, και με βάση το αρχικό μοντέλο εφαρμόζουμε ένα halftone effect για καλύτερη οπτικοποίηση της σκηνής

Το τελικό αποτέλεσμα είναι το εξής:



(α) Το κανονικό δωμάτιο Peirene Fountain.



(β) Το δεύτερο effect μας εφαρμοσμένο στο Peirene Fountain.

Σχήμα 3.29: Η επίπτωση εφαρμογής του εφέ μας στον δεύτερο πίνακα-δωμάτιο.



Σχήμα 3.30: Άλλη όψη του εφέ.

Επίσης, μπορούμε να παρατηρήσουμε το εφέ μέσα από τον πίνακα του κεντρικού Studio Room.

### c. Brush Stroke effect

Για το τρίτο δωμάτιο ζητείται να αναπτύξουμε ένα brush stroke effect, με καθαρό στόχο να πετύχουμε τα οπτικά αποτελέσματα ενός πίνακα ζωγραφισμένο με πινέλο. Αν και υπάρχουν περίπλοκοι αλγόριθμοι που μπορούν να πετύχουν αυτόν τον στόχο που εξετάσαμε, τελικά καταλήξαμε σε ένα φίλτρο που μπορεί να εφαρμοστεί σε κώδικα των shader: το φίλτρο Kuwahara.

Το φίλτρο Kuwahara αναπτύχθηκε αρχικά για απο-θορυβοποίηση ιατρικών απεικονίσεων, αφού σε αντίθεση με άλλα de-noising φίλτρα, διατηρεί ακμές. Το αρτίστικο painting effect που προκαλεί για τεχνικούς σκοπούς ήταν ανεπιθύμητο, όμως για μας και για image editing είναι επιθυμητό αποτέλεσμα. Σε γενικές γραμμές, ο αλγόριθμος είναι ο εξής:

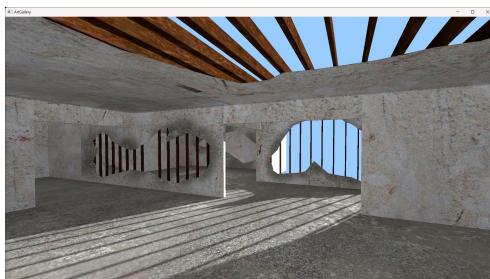
- Ορίζουμε το μέγεθος του παραθύρου με το οποίο θα εξετάζουμε γειτονικά πίξελ.
- Διατρέχουμε όλα τα πίξελ της εικονάς.
- Χωρίζουμε σύμφωνα με το παράθυρο τα πίξελ σε 4 υπο-περιοχές.
- Για κάθε υπο-περιοχή υπολογίζουμε τη μέση τιμή και την διασπορά
- Επιλέγουμε την υπο-περιοχή με την ελάχιστη διασπορά, που επιδεικνύει ελάχιστη αλαγή κατά το texture, και θέτουμε τη καινούργια τιμή του πίξελ στη μέση τιμή της επιλεγμένης περιοχής
- Συνεχίζουμε για όλα τα πίξελ

Τα αποτελέσματα για το Destroyed Room είναι τα εξής:

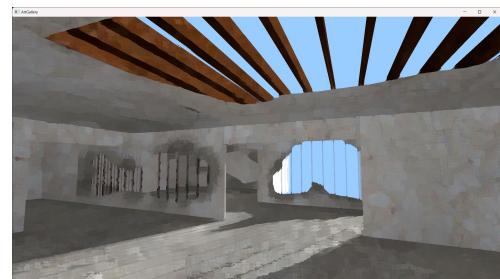


Σχήμα 3.31: Εφαρμογή του Kuwahara filter στο δωμάτιο Destroyed Room.

Για καλύτερη παρουσίαση του εφέ, συγκρίνουμε την αρχική εικόνα και μετά με το φίλτρο ενεργοποιημένο:



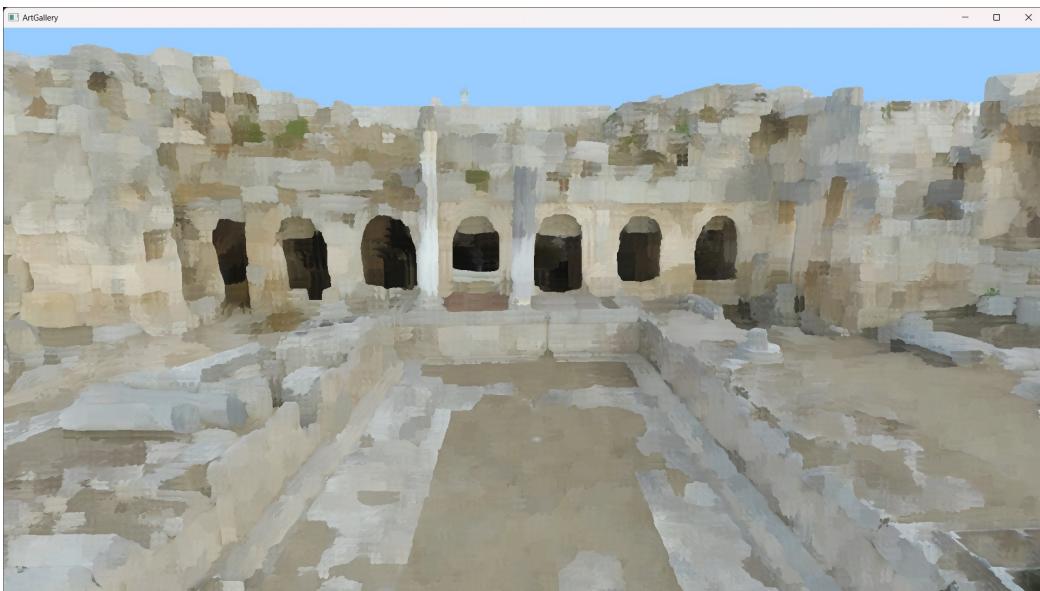
(α) Το κανονικό δωμάτιο Destroyed Room.



(β) Το τρίτο effect μας εφαρμοσμένο στο Destroyed Room.

Σχήμα 3.32: Η επίπτωση εφαρμογής του εφέ μας στον τρίτο πίνακα-δωμάτιο.

Και εφαρμοσμένο στο δωμάτιο Peirene Fountain:



Σχήμα 3.33: Εφαρμογή του Kuwahara filter στο δωμάτιο Peirene Fountain.

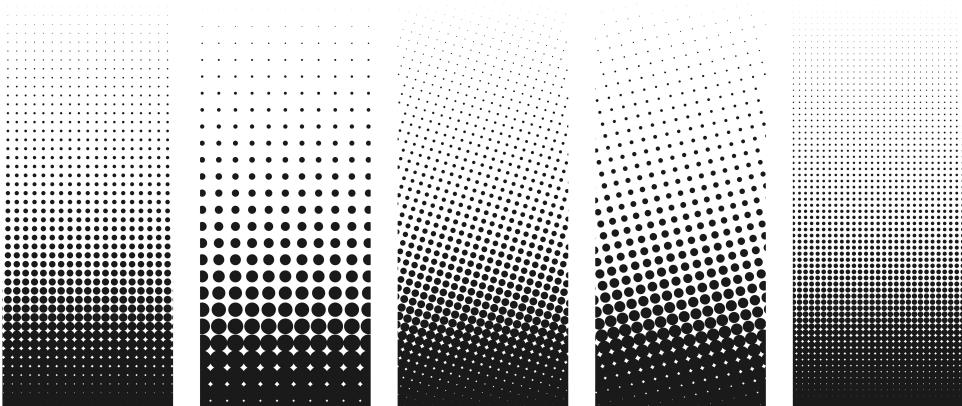
Για να ενισχύσουμε το painting effect του φίλτρου, χρησιμοποιούμε μεγάλο kernel size επίτηδες.

#### d. Comic Book Effect

Στο τέταρτο δωμάτιο μάς ζητείται η εφαρμογή ενός grayscale comic book effect, όμως για λόγους μοναδικότητας και για να ξεχωρίζουν καλύτερα οι πίνακες μεταξύ τους, θα το εφαρμόσουμε έγχρωμα.

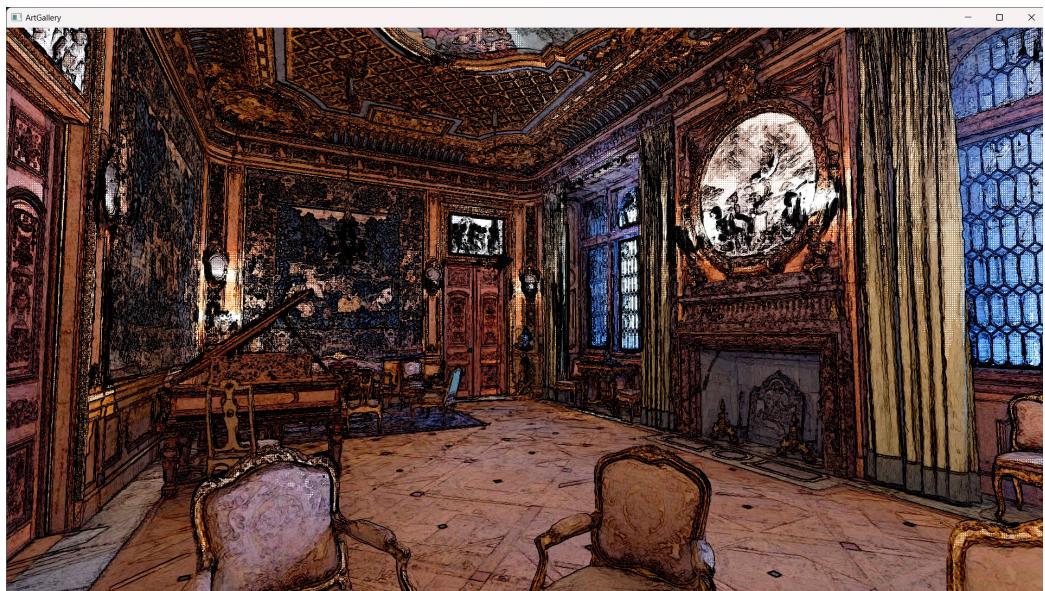
Υπάρχουν αρκετές εφαρμογές cell-shading που φέρνουν μια αίσθηση comic book, όμως θα αναπτύξουμε τη δική μας προσέγγιση μέσω από φίλτρα στους shaders.

Τα χαρακτηριστικά του στύλ των comic books είναι έντονα χρώματα, έντονες σιλουέτες και outlines, το διαδεδομένο halftone effect (που παρουσιάζεται παρακάτω) και έντονες, στυλιστικές μαύρες γραμμές σχεδιαγραμμάτων.



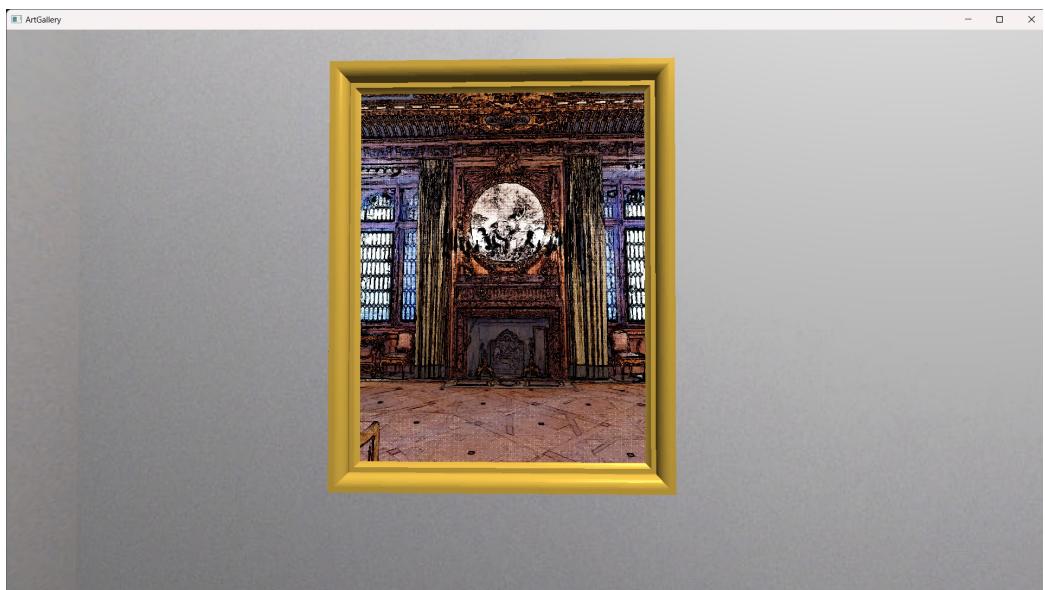
Σχήμα 3.34: To halftone effect, διαδεδομένο στην εκτύπωση των comic books. Ανεβασμένο από χρήστη starline, “Abstract grunge halftone distorted shapes background design”, [freepik.com](https://www.freepik.com).

Επομένως, υψώνουμε το contrast των χρωμάτων του texture, εφαρμόζουμε ένα Sobel φίλτρο [4.4] για ανίχνευση και ύστερη ενίσχυση των περιγραμμάτων των αντικειμένων, και το halftone effect που συζητήσαμε νωρίτερα για τον πουαντιγισμό. Ο συνδυασμός όλων αυτών των φίλτρων επιφέρει το ακόλουθο αποτέλεσμα στον πίνακα-δωμάτιο Great Drawing Room:



Σχήμα 3.35: Το comic book effect εφαρμοσμένο πάνω στο μοντέλο Great Drawing Room.

Μπορούμε επιπλέον να παρατηρήσουμε το εφέ, όπως συνήθως, κι από το Studio Room.

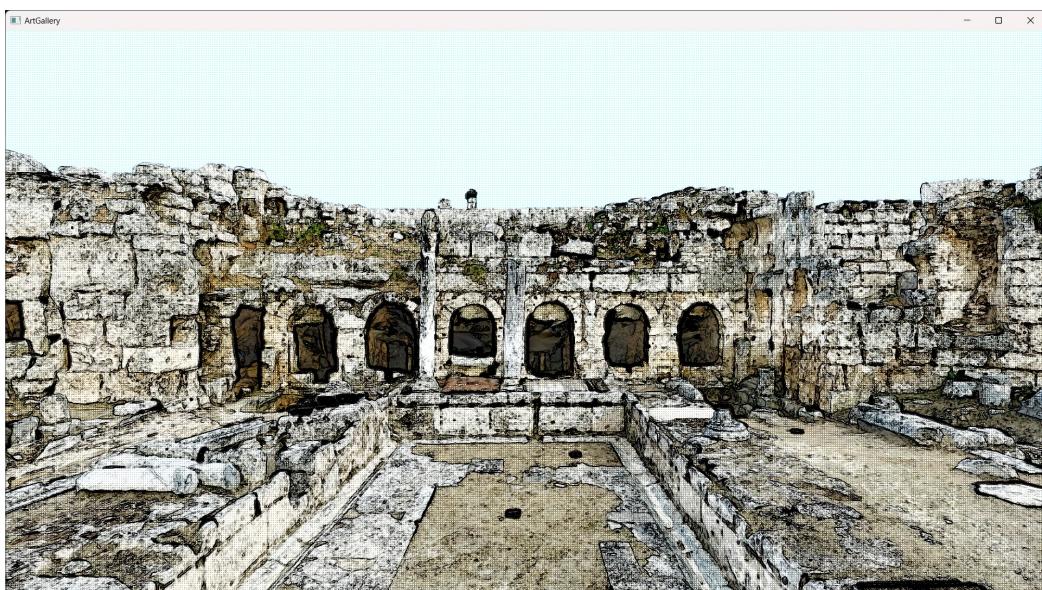


Σχήμα 3.36: Το comic book effect εφαρμοσμένο πάνω στο portal για τον πίνακα του Μεγάλου Δωματίου Ζωγραφικής.

Και για λόγους σύγκρισης εφαρμόζουμε και το εφέ σε άλλα δωμάτια:



Σχήμα 3.37: Το comic book effect εφαρμοσμένο πάνω στο δωμάτιο Morning Room.



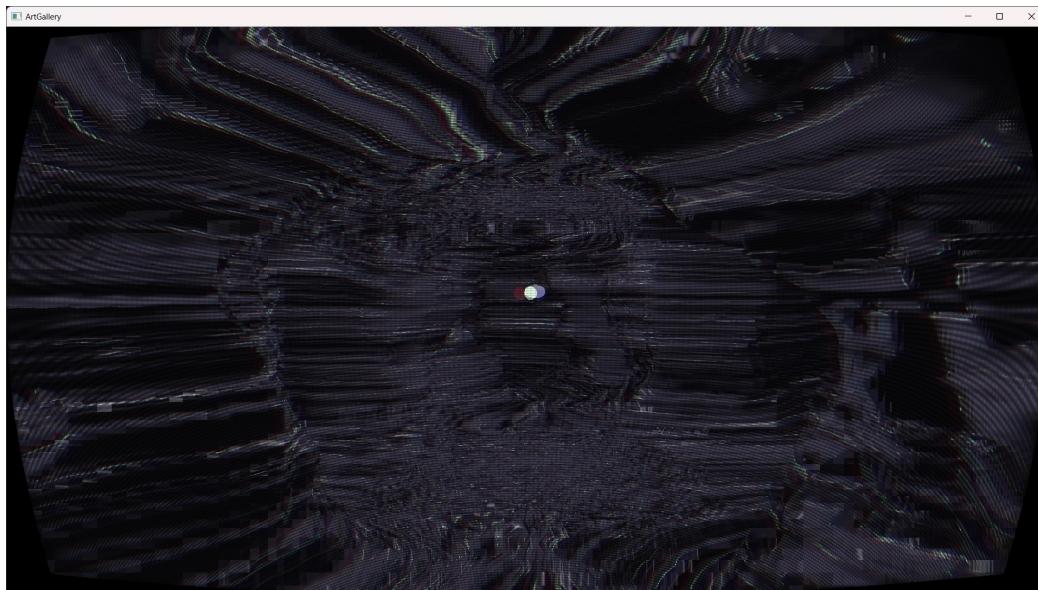
Σχήμα 3.38: Το comic book effect εφαρμοσμένο πάνω στο δωμάτιο Peirene Fountain.

### e. CRT effect

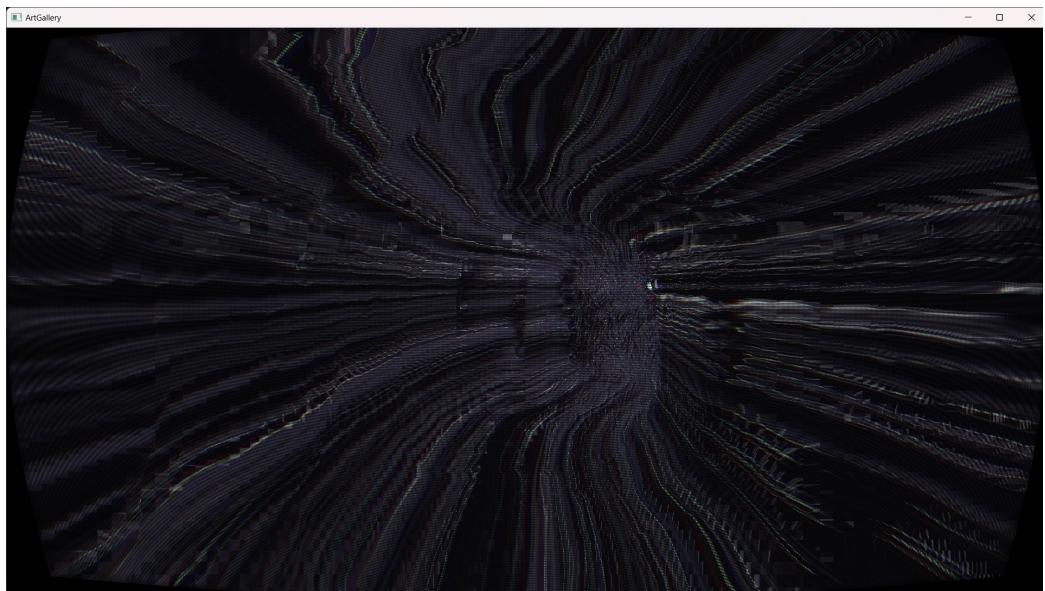
Μας ζητείται να αναπτύξουμε το δικό μας στυλ για το τελευταίο πέμπτο δωμάτιο. Θα αναπτύξουμε ένα CRT effect εφέ τηλεόρασης, το οποίο θα εφαρμοστεί στη Μεταλλική Σπηλιά.

Τα χαρακτηριστικά ενός CRT effect είναι μια βινιέτα, ένα πλαίσιο που ακυρώνει τις ορθές γωνίες των σημερινών οθόνων, καμπυλότητα όπως το γυαλί των τηλεοράσεων καθοδικών σωληνών, ατέλειες και artifacts στη μετάδοση του σήματος καθώς και εμφάνιση του φαινομένου χρωματικού σφάλματος (chromatic aberration).

Όλως ο κώδικας δύναται να πραγματοποιηθεί στους shaders, όπου και τον αναπτύσσουμε με τα ακόλουθα στατικά αποτελέσματα για την εικόνα:

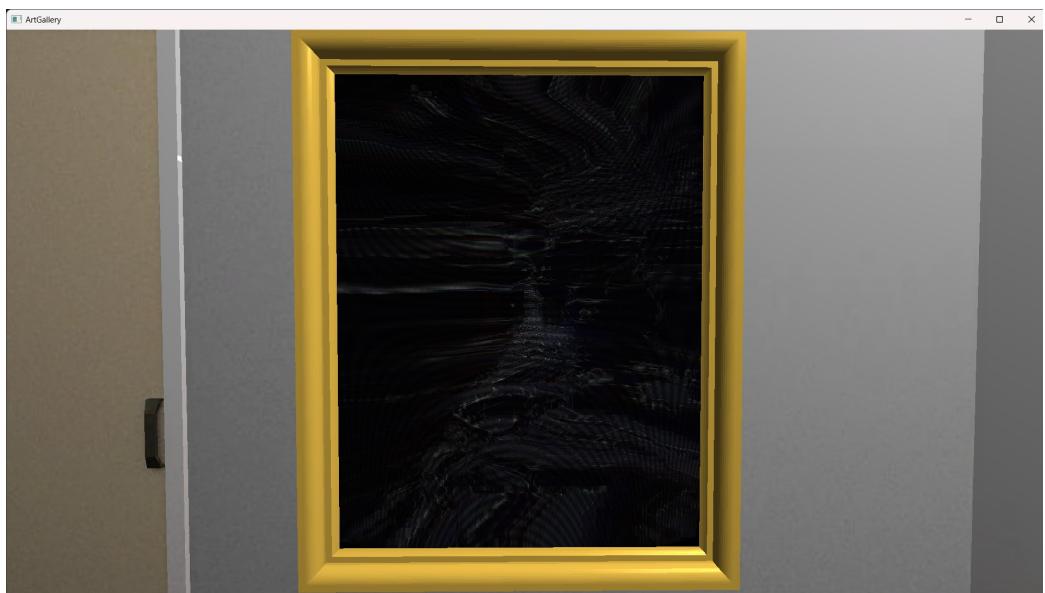


Σχήμα 3.39: Το CRT effect εφαρμοσμένο στο δωμάτιο του Metallic Cave.



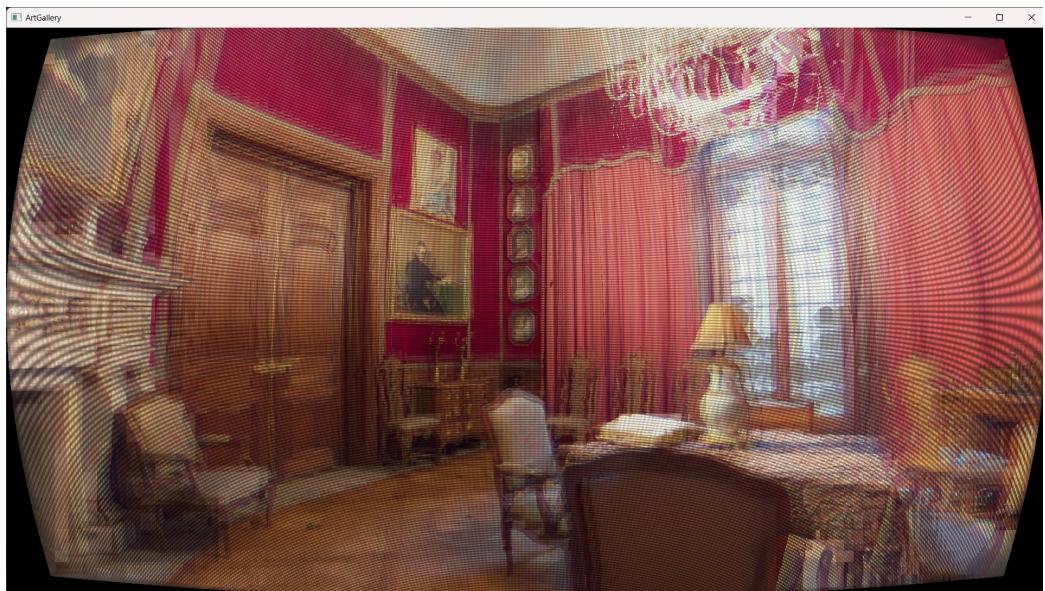
Σχήμα 3.40: Άλλη όψη του εφέ για το Metallic Cave δωμάτιο.

Αν εφαρμόσουμε το εφέ αυτό στο portal του κεντρικού δωματίου, έχουμε τη παρακάτω σκηνή:



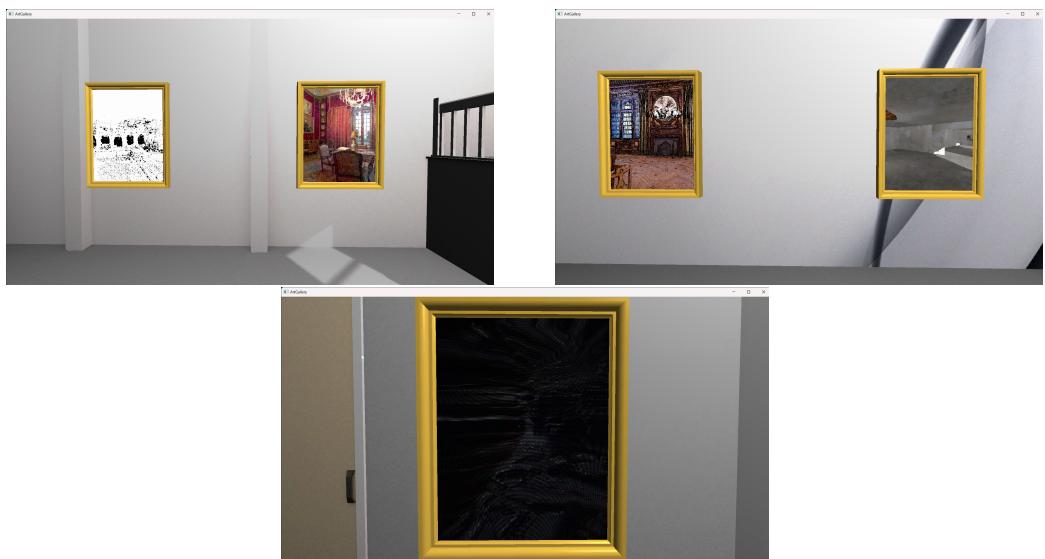
Σχήμα 3.41: To CRT effect εφαρμοσμένο πάνω στο portal για τον πίνακα του Metallic Cave.

Και σε διαφορετικό δωμάτιο από τη Μεταλλική Σπηλιά:



Σχήμα 3.42: To CRT effect εφαρμοσμένο στο δωμάτιο του Morning Room.

Και ολοκληρώνουμε την εφαρμογή των 5 διαφορετικών στυλ στους 5 πίνακες-δωμάτιά μας.



Σχήμα 3.43: Τα 5 εφέ εφαρμοσμένα στους πίνακες-δωμάτια στο κεντρικό Studio Room.

### 3.2.2 Ερώτημα 5

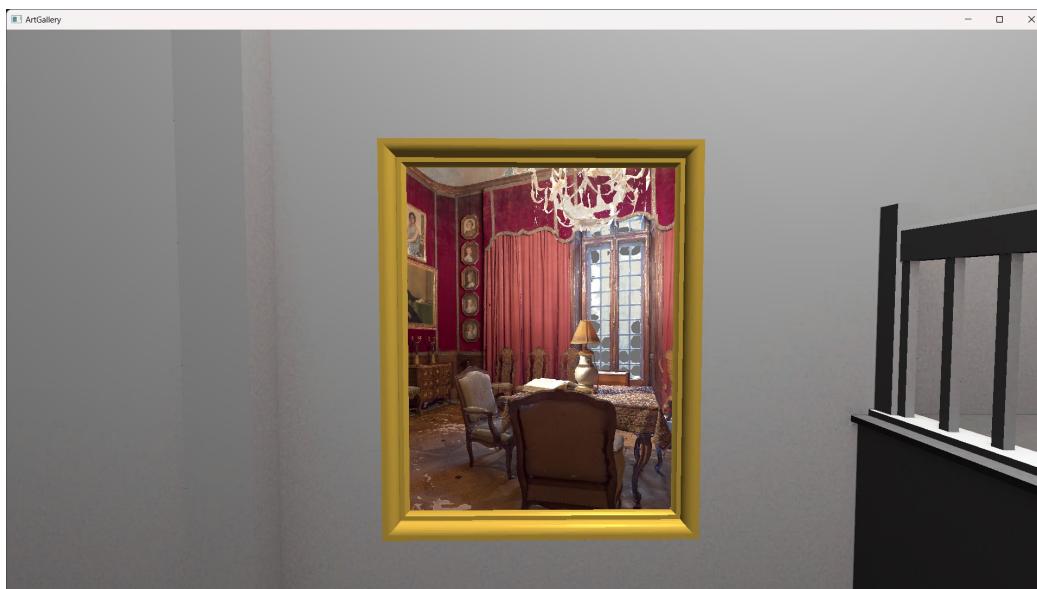
Το επόμενο βήμα της εργασίας είναι να δημιουργήσουμε Απεικόνιση Αναγλύφου (bump map) των πινάκων στο Studio Room σύμφωνα με τα κανονικά διανύσματα (normal vectors) των δωματίων.

Καθίσταται προφανές το γεγονός ότι η ευκολότερη προσέγγιση είναι αυτή που έχουμε ακολουθήσει ως το εξής: Το rendering της σκηνής σε ιδιαίτερο framebuffer, αποθήκευση της πληροφορίας που μας ενδιαφέρει στο attached texture και ύστερα δειγματοληψία στους τελικούς shaders που πραγματοποιούν την απεικόνιση στο quad της οθόνης.

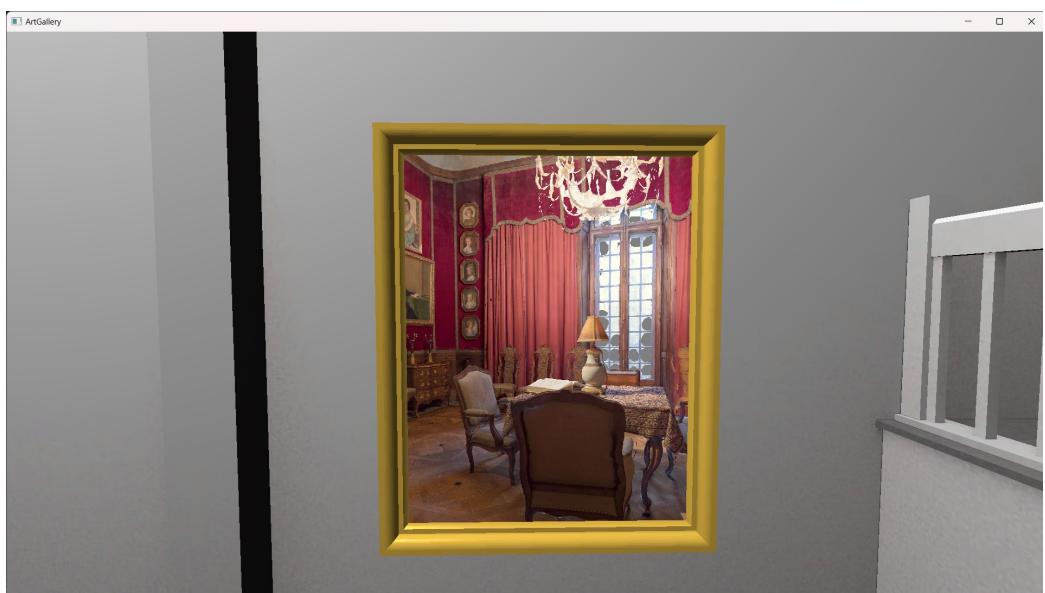
Ίδια τακτική υιοθετούμε και για τη παραγωγή χαρτών των normal. Μαζί με το portal pass για τη παραγωγή χρωματισμένων texture που αποτελούν το “παράθυρο” στα άλλα δωμάτια, θα πραγματοποιήσουμε και ένα επιπλέον pass για παραγωγή του αντίστοιχου texture με αποθηκευμένα τα normals.

Εφόσον περάσουμε το normal map στους shaders, μπορούμε να κάνουμε sample το κανονικό διάνυσμα και να το χρησιμοποιήσουμε στους υπολογισμούς φωτισμού Phong αργότερα.

Το αποτέλεσμα στους πίνακες είναι καλύτερα παρατηρίσιμο όταν το φως βρίσκεται ύπο κίνηση, όμως μερικές εικόνες από την εφαρμογή του bump mapping φαίνονται παρακάτω.

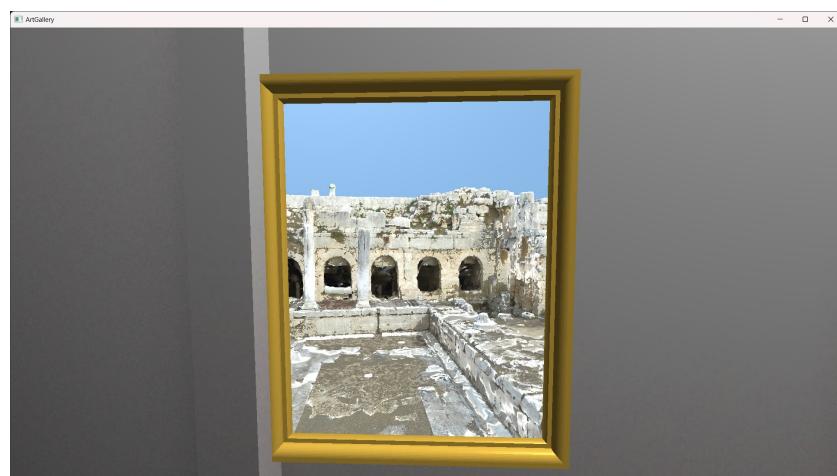


Σχήμα 3.44: Ο πίνακας του Morning Room με normal mapping.

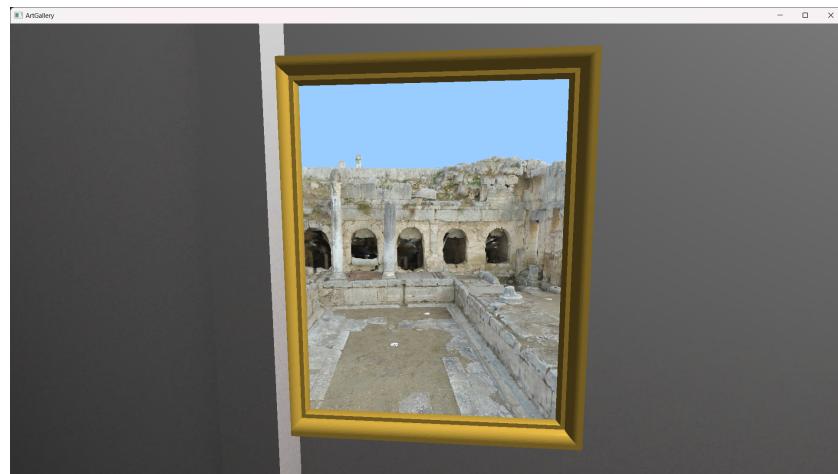


Σχήμα 3.45: Ο αρχικός πίνακας του Morning Room χωρίς normal mapping.

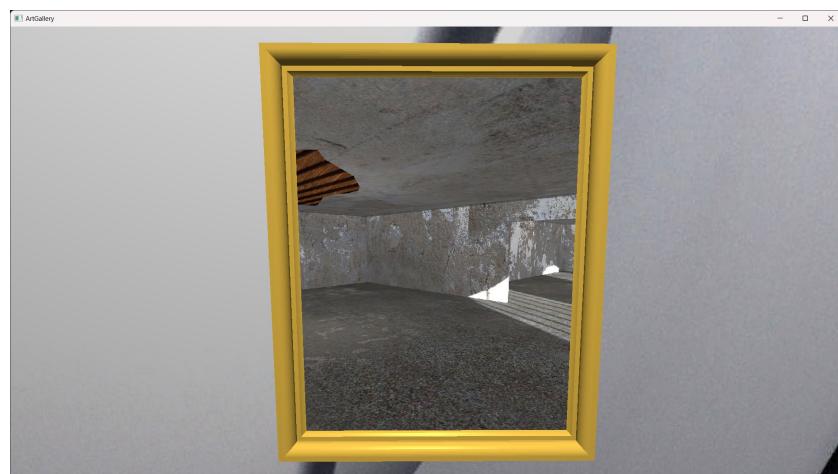
Οι διαφορές είναι ελαφρές και δυσεύρετες από εικόνα, αλλά υπαρκτές. Για λόγους πληρότητες, παραθέτουμε τις ίδιες φωτογραφίες για όλους τους πίνακες:



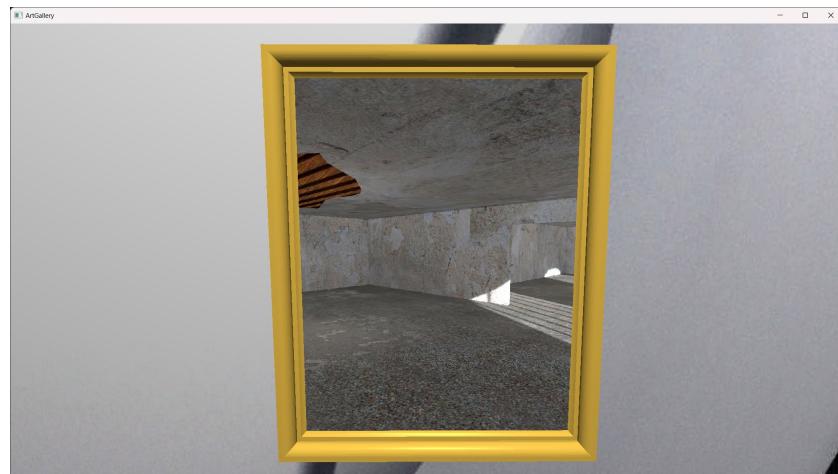
Σχήμα 3.46: Ο πίνακας του Peirene Fountain με normal mapping.



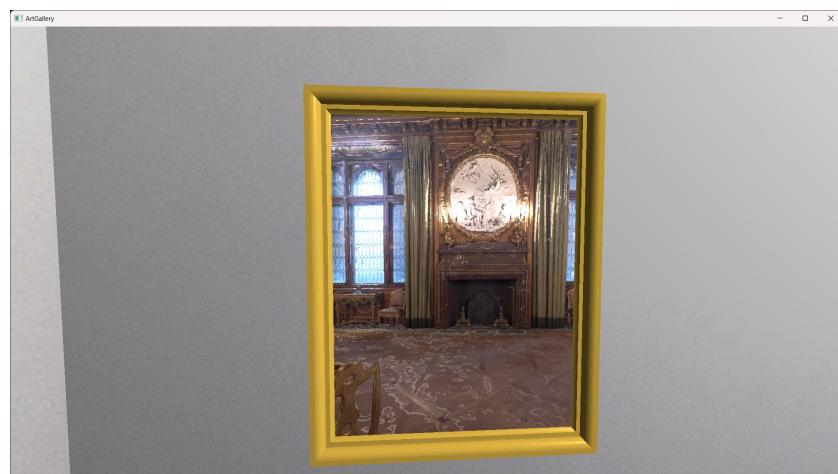
Σχήμα 3.47: Ο αρχικός πίνακας του Peirene Fountain χωρίς normal mapping.



Σχήμα 3.48: Ο πίνακας του Destroyed Room με normal mapping.



Σχήμα 3.49: Ο αρχικός πίνακας του Destroyed Room χωρίς normal mapping.



Σχήμα 3.50: Ο πίνακας του Great Drawing Room με normal mapping.

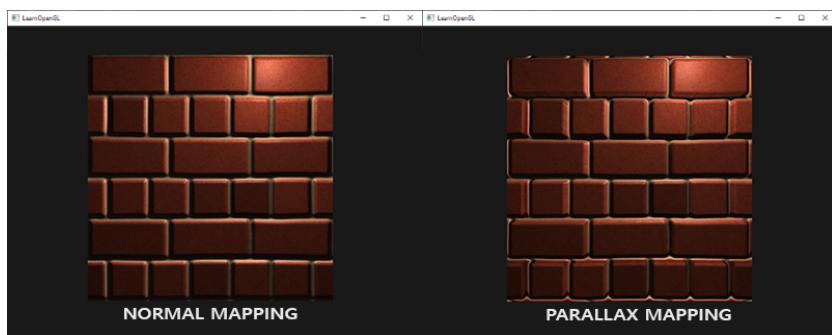


Σχήμα 3.51: Ο αρχικός πίνακας του Great Drawing Room χωρίς normal mapping.

### 3.2.3 Ερώτημα 6

Τέλος, το έκτο ερώτημα της εργασίας ζητάει, με τη χρήση ενός χάρτη βάθους σκηνής (depth map) να ανυψώσουμε το mesh των πινάκων για ανάγλυφη αναπαράσταση της σκηνής.

Μια συμπληρωματική τεχνική σε αυτή του bump mapping είναι το parallax mapping [4.2]. Το parallax mapping προσπαθεί να προσομοιώσει εικονικά το βάθος, βασιζόμενο σε πληροφορίες όπως το επιθυμητό βάθος από τον χάρτη βάθους σκηνής και τη τοποθεσία από την οποία παρατηρούμε το πίξελ (ή fragment). Επιτυχαίνει εντυπωσιακά αποτελέσματα χωρίς να βαραίνει το πρόγραμμα με πάρα πολλά επιπλέον vertices.



Σχήμα 3.52: Εφαρμογή του parallax mapping μαζί με normal mapping, φωτογραφεία του [learnopengl.com](http://learnopengl.com).

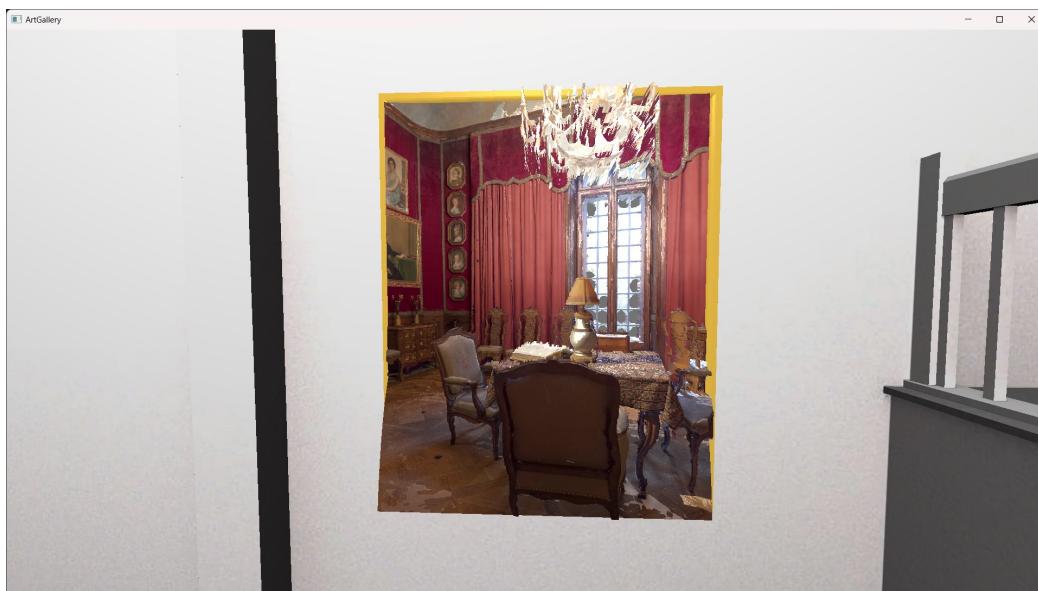
Η εφαρμογή της τεχνικής στην σκηνή μας, δεν επέφερε σημαντικές αλλαγές, είτε λόγω κωδικοποίησεις του depth map, σφάλματος στην εφαρμογή ή ακαταλ-

ληλότητας της σκηνής. Επομένως, θα υλοποιήσουμε πραγματική ανύψωση των vertices, ως εναλλακτική του parallax mapping.

Πρώτα όμως, πρέπει να κατασκευάσουμε τους χάρτες βάθους από το rendering των σκηνών. Ευτυχώς για μας, για κατάλληλη κατασκευή των χρωματισμένων texture των portals, υποχρεωτικά απαιτείται depth component attachment ώστε να μπορούν να γίνουν τα vertices που βρίσκονται πιο “μπροστά” από άλλα rendered σωστά, δηλαδή να κάνουν overwrite τα μπροστά fragments τα πισινά fragments. Επομένως έχουμε ήδη πρόσβαση στα depth maps των δωματίων.

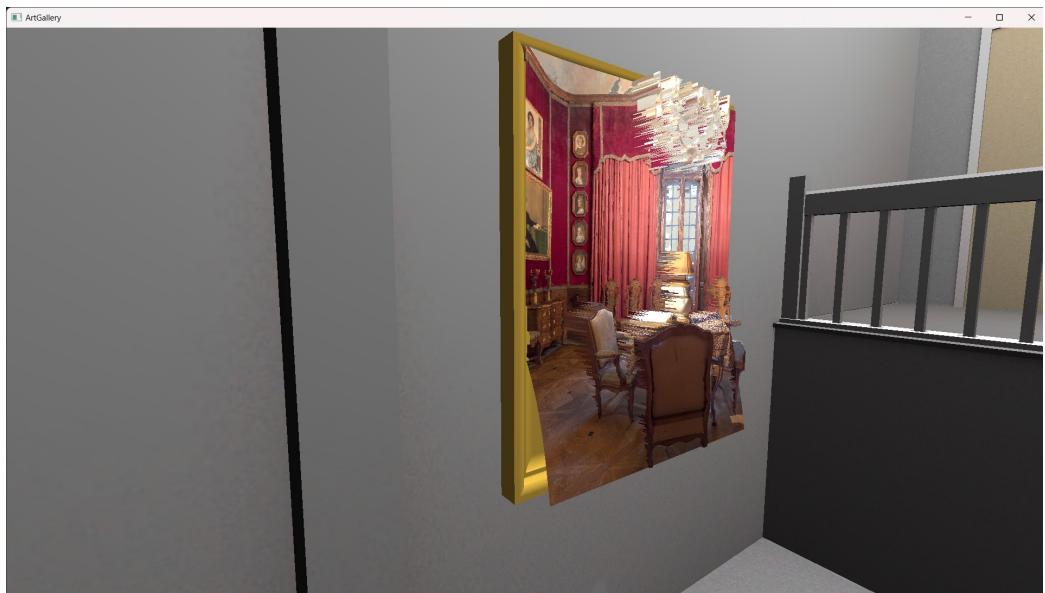
Για τα post-processing effects των προηγούμενων ερωτημάτων, καθώς και για το normal mapping και parallax mapping, δουλεύαμε στον fragment shader. Τώρα όμως, θα επεκτείνουμε το μοντέλου του ορθογωνίου στο οποίο ζωγραφίζεται το texture του portal με περισσότερα από 4 vertices, έστω σε ένα grid  $N \times N$ , και θα επέμβουμε στον vertex shader, όπου και θα αλλάξουμε την world-space τοποθεσία των vertex, σύμφωνα με δειγματοληψία από το depth map.

Επομένως λαμβάνουμε τα παρακάτω αποτελέσματα:



Σχήμα 3.53: Ανύψωση του mesh σύμφωνα με το depth map του δωματίου του Morning Room.

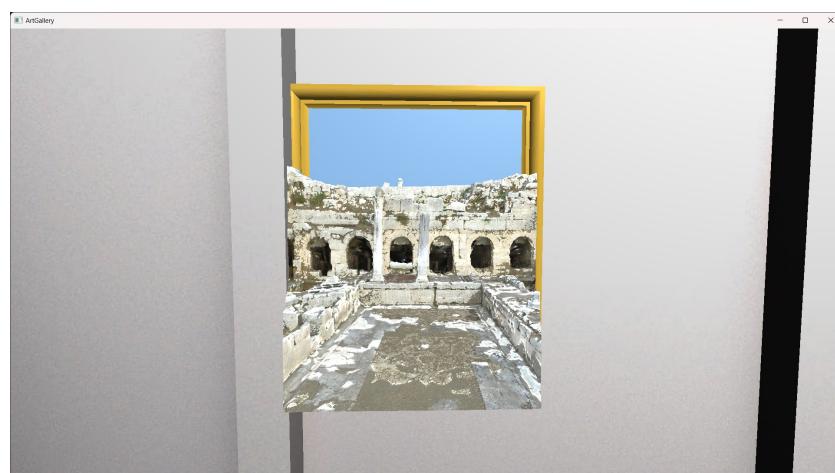
Αν παρατηρήσουμε τον πίνακα από τα πλάγια, μπορούμε να δούμε τις παραμορφώσεις που οφείλονται στην ανύψωση.



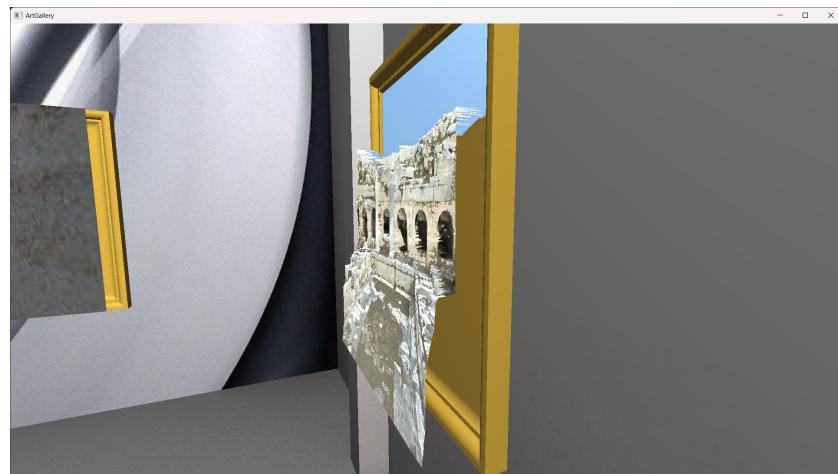
Σχήμα 3.54: Πλαϊνή όψη του ανυψωμένου πίνακα του Morning Room.

Προφανώς, το texture δεν περιέχει πληροφορία για την υπό γωνία προβολή του ανυψωμένου mesh.

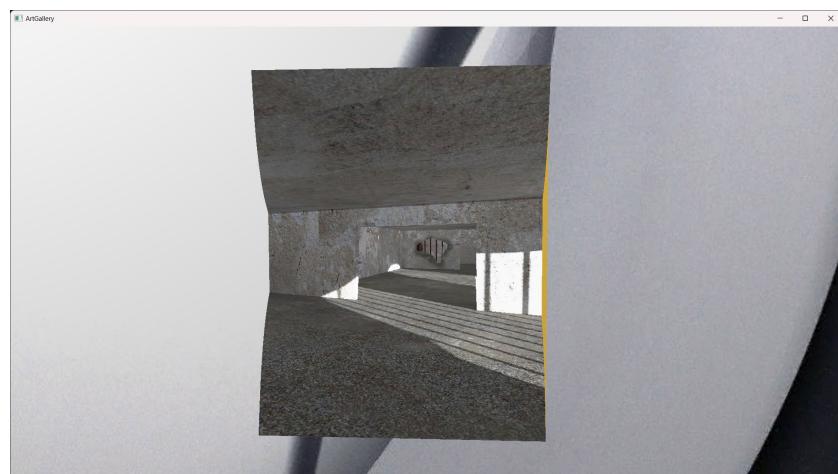
Για λόγους πληρότητας, παραθέτουμε και τους υπόλοιπους πίνακες, εκτός του Metallic Cave, το οποίο είναι το πιο δύσκολο να προβληθεί μέσα από στατικές εικόνες:



Σχήμα 3.55: Ανύψωση του mesh σύμφωνα με το depth map του δωματίου του Peirene Fountain.



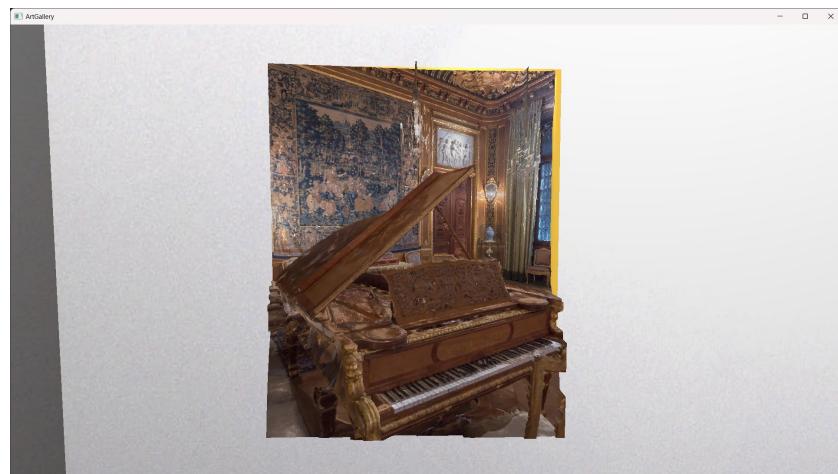
Σχήμα 3.56: Πλαϊνή όψη του ανυψωμένου πίνακα του Peirene Fountain.



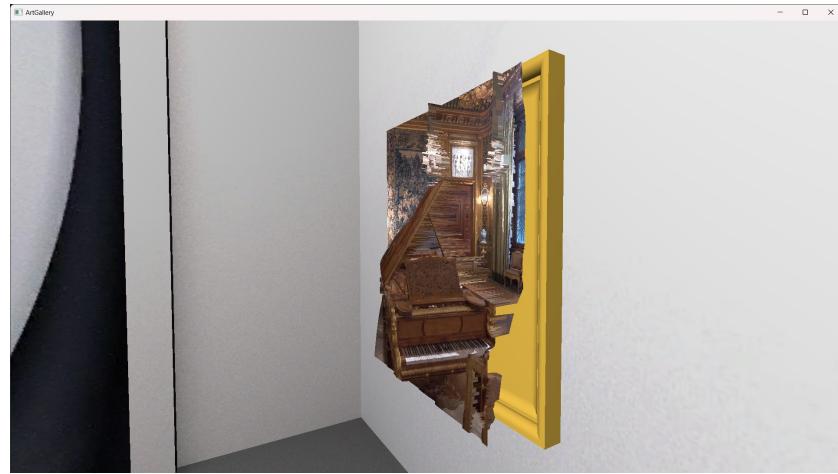
Σχήμα 3.57: Ανύψωση του mesh σύμφωνα με το depth map του δωματίου του Destroyed Room.



Σχήμα 3.58: Πλαϊνή όψη του ανυψωμένου πίνακα του Destroyed Room.

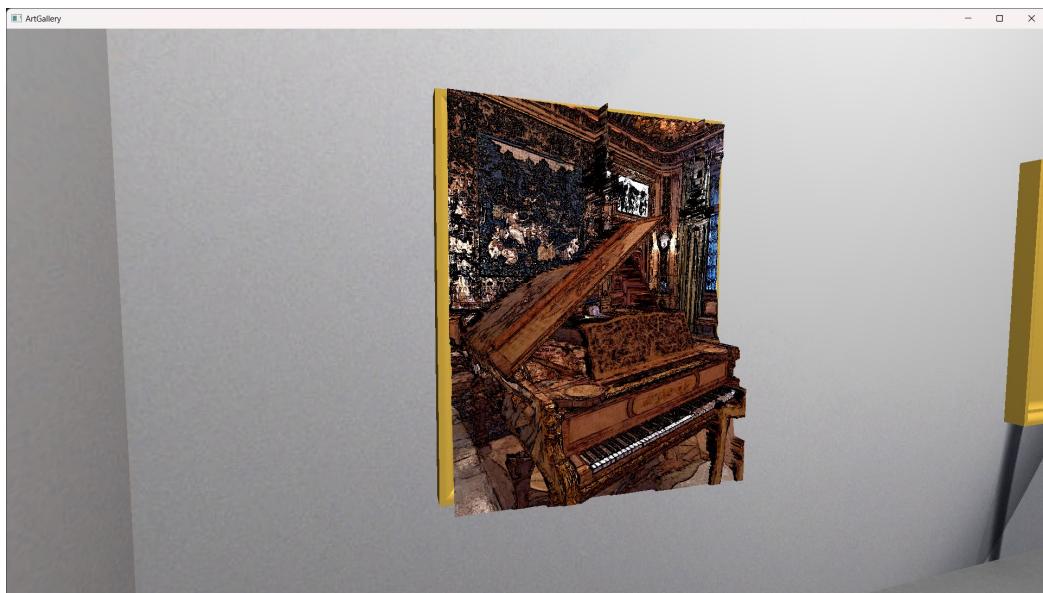


Σχήμα 3.59: Ανύψωση του mesh σύμφωνα με το depth map του δωματίου του Great Drawing Room.



Σχήμα 3.60: Πλαϊνή όψη του ανυψωμένου πίνακα του Great Drawing Room.

Επίσης, η ανύψωση του mesh δε μας απαγορεύει από την εφαρμογή του εφέ στον πίνακα, και μπορούμε να λάβουμε εντυπωσιακά αποτελέσματα:



Σχήμα 3.61: Ανύψωση του mesh σύμφωνα με το depth map του δωματίου του Great Drawing Room και εφαρμογή του comic book effect.

### 3.2.4 Ερώτημα Extra - Transitions

Μέχρι τώρα, η χρήση του προγράμματος και η πρόσβαση στα υπόλοιπα δωμάτια έχει πραγματοποιηθεί στατικά. Ο χρήστης πλησιάζει τον πίνακα, αν βρίσκεται

επαρκώς κοντά και πατήσει SPACE τηλεμεταφέρεται εντός του πίνακα-δωματίου. Αν το ξαναπατήσει επιστρέφει στο Studio Room.

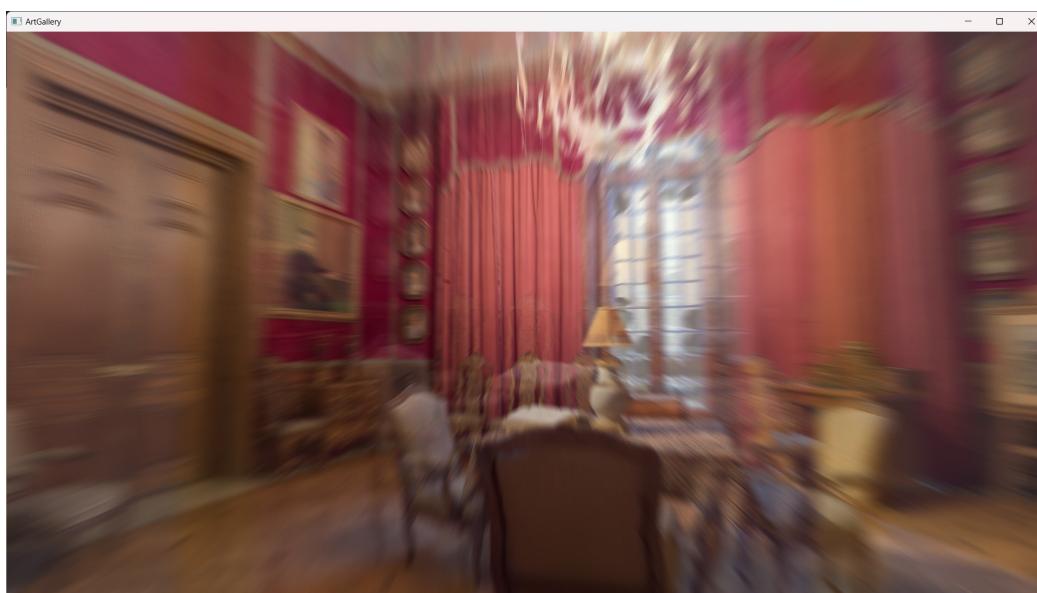
Στο παρόν ερώτημα επιχειρούμε να πραγματοποιήσουμε transitions της σκηνής, από την απεικόνιση στο Studio Room σε κάποιο άλλο δωμάτιο και αντίστροφα.

Για να το πραγματοποιήσουμε αυτό, πρέπει να κάνουμε hijack ολόκληρου του main loop εφόσον βρισκόμαστε σε κατάσταση transitioning, να εντοπίσουμε αν τηλεμεταφερόμαστε από το κεντρικό δωμάτιο σε δωμάτιο-πίνακα ή αντιστρόφως και να εφαρμόσουμε το επιλεγμένο mixing και blending. Οι αλλαγές είναι πλήρως “εικονικές” και εφαρμόζονται στα displayed textures, οπότε μόνο χρειάζεται να γράψουμε κώδικα στον fragment shader.

Είναι δύσκολη η απεικόνιση των transition σε μορφή εικόνας, λόγω της φύσεως του ως χρονικό συμβάν, όμως μερικές ενδεικτικές εικόνες παρουσιάζονται παρακάτω, μαζί με σύντομη περιγραφή της μετάβασης (transition).

### Transition 1 - Blurring Crossfade

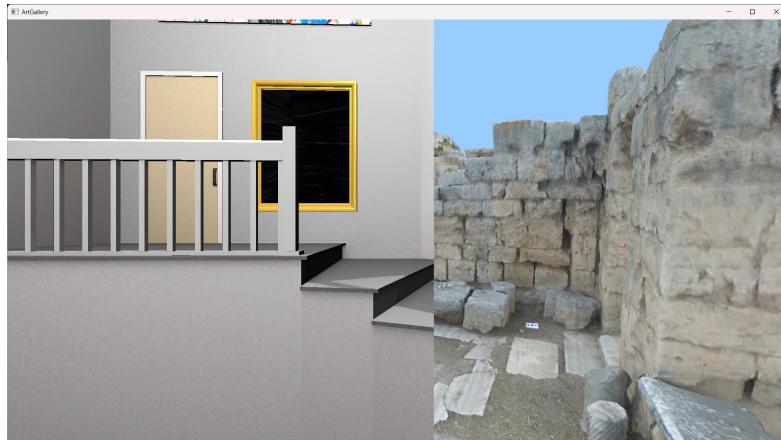
Η πρώτη μετάβαση είναι τύπου crossfade, όπου οι opacity τιμές των δύο texture μειώνονται αναλόγως, μόνο ενισχυμένο με τεχνικές για εκθετικό/ημιτονοειδές ease in-out και blurring του περίγυρου για ενισχυμένη αίσθηση της μετάβασης/τηλεμεταφοράς.



Σχήμα 3.62: Μετάβαση από το κεντρικό δωμάτιο προς το Morning Room με το transition 1.

## Transition 2 - Horizontal Swipe

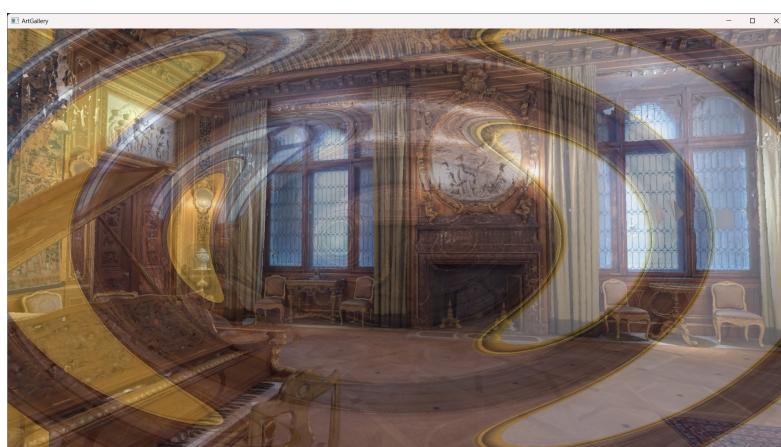
Σαν καρέ που από τα αριστέρα προς τα δεξιά επικαλύπτει το προηγούμενο, εφαρμόζουμε τη μετάβαση αυτή, η οποία έχει διακριτή αρχή και τέλος για τα δύο textures.



Σχήμα 3.63: Μετάβαση από το Peirene Fountain προς το κεντρικό δωμάτιο με το transition 2.

## Transition 3 - Water Drop

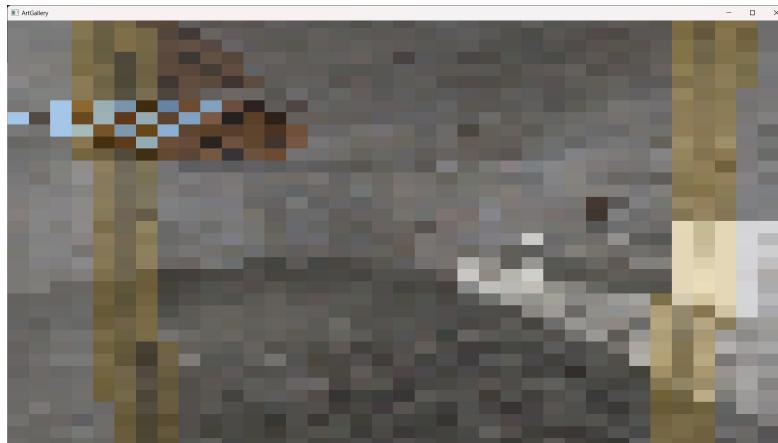
Στο transition αυτό, χρησιμοποιούνται ημίτονα για παραγωγή παραμορφώσεων στην οθόνη σαν κυματισμούς σε σώμα νερού. Το κύμα, όπως συνήθως, καθορίζεται από τη ταχύτητα διάδοσής του και της ισχύς του.



Σχήμα 3.64: Μετάβαση από το Studio Room προς το Great Drawing Room με το transition 3.

### **Transition 4 - Pixelated Dissolve**

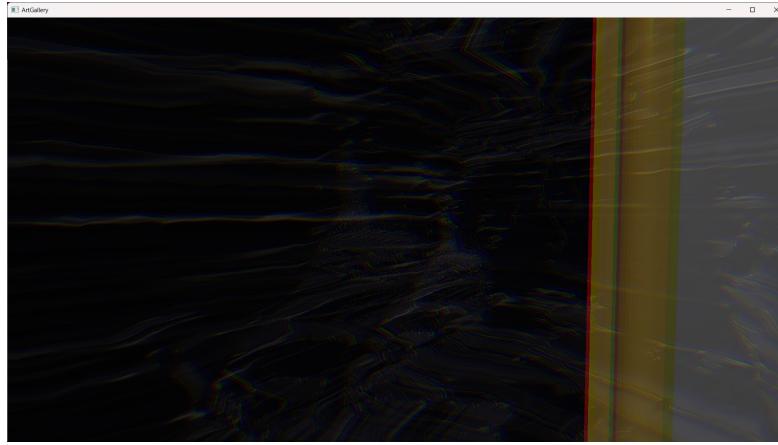
Η μετάβαση αυτή περιλαμβάνει ορισμό ενός μεγέθους, το οποίο καθορίζει το pixelization στην εικόνα, όσο ο χρόνος του transition αυξάνεται, το μέγεθος ενός “πίξελ” αυξάνεται, το πλήθος τους που αποτελούν την εικόνα μειώνεται, μέχρι που θολώνει εντελώς και μεταβαίνουμε στην επόμενη σκηνή.



Σχήμα 3.65: Μετάβαση από το Studio Room προς το Destroyed Room με το transition 4.

### **Transition 5 - Glitched Memories**

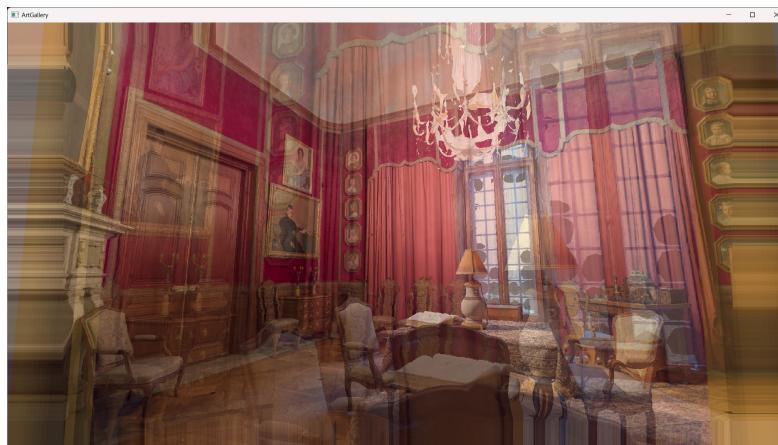
Στη μετάβαση 5, προσθέτουμε τυχαίο θόρυβο στα 3 κανάλια χρώματος RGB που μεταβάλλεται σύμφωνα με τον χρόνο, και καταλήγουμε στο αποτέλεσμα μιας οθόνης που “δονείζεται” έντονα, με chromatic aberration και δημιουργεί την εντύπωση σφάλματος (glitch) στην σκηνή



Σχήμα 3.66: Μετάβαση από το Studio Room προς τη Μεταλλική Σπηλιά με το transition 5.

### Transition 6 - Zoom Blur Transition

Παρόμοιο με το transition 1, αλλά λιγότερο smooth, στη μετάβαση αυτή το δωμάτιο προς το οποίο τηλεμεταφερόμαστε έρχεται και τοποθετείται στα όρια της οθόνης από απόσταση, που δίνει την αίσθηση ότι πέφτουμε μέσα του.



Σχήμα 3.67: Μετάβαση από το κεντρικό δωμάτιο προς το Morning Room με το transition 6.

### Transition 7 - Radial Smooth Wipe

Όμοιας λογικής με το transition 2, όμως η μετάβαση αυτή πραγματοποιείται ομαλά και ακτινικά, από το κέντρο της εικόνας προς το εξωτερικό.

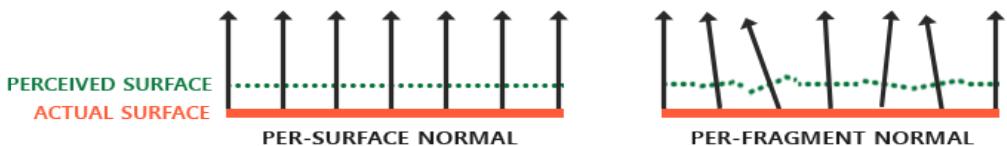


Σχήμα 3.68: Μετάβαση από το Peirene Fountain προς το Studio Room με το transition 7.

# 4 Θεωρία/Θεωρητικό Υπόβαθρο

## 4.1 Normal Mapping

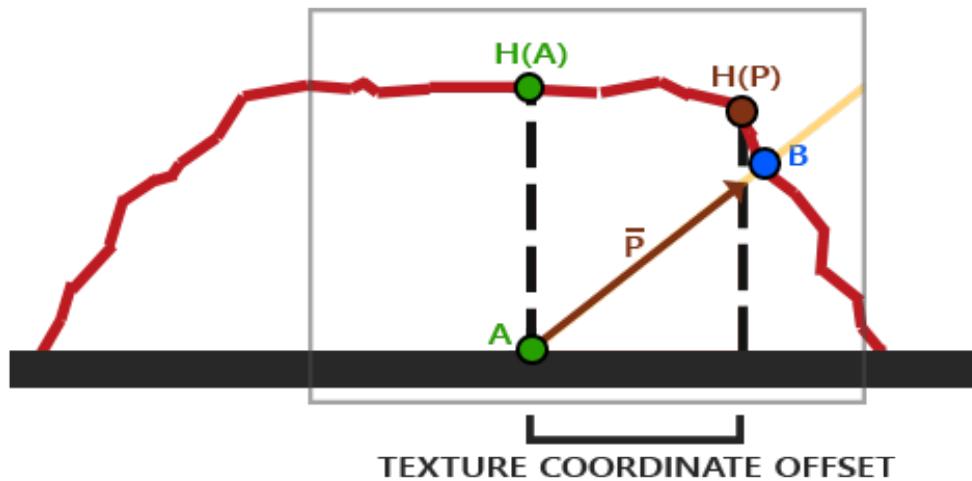
Το Normal mapping ενισχύει λεπτομέρειες μιας επιφάνειας, χρησιμοποιώντας ένα τρίτο texture που περιέχει τις πληροφορίες κανονικού διανύσματος. Αυτό επιτρέπει πιο ρεαλιστικούς υπολογισμούς φωτισμού, προσομοιώνοντας “σκαμπανεβάσματα” και βαθουλώματα, χωρίς περαιτέρω περιπλοκη γεωμετρία. Ο χάρτης κανονικών διανυσμάτων συνήθως υπολογίζεται σε tangent space και το μεταμορφωμένο normal χρησιμοποιείται ύστερα. Έτσι επιτυγχάνουμε περιπλοκες επιφάνειες χωρίς η πολυπλοκότητα των μοντέλων μας να ανεβαίνει



Σχήμα 4.1: Η λογική του normal mapping για κάθε fragment.

## 4.2 Parallax Mapping

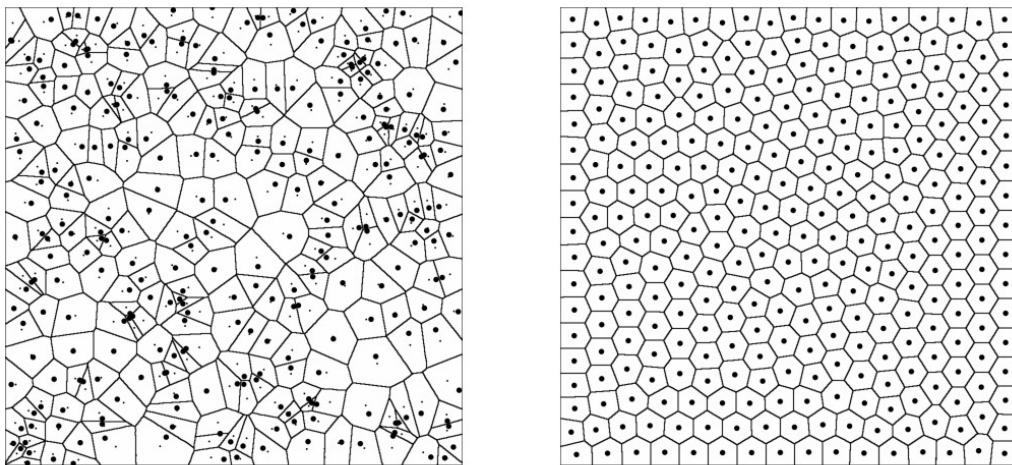
Το Parallax mapping, παρομοίως με το normal mapping [4.1], ενισχύει την αίσθηση του βάθους σε μια επιφάνεια μετατοπίζοντας τις συντεταγμένες του texture σύμφωνα με την γωνία προβολής. Έτσι, προσομοιώνομε parallax occlusion, δημιουργόντας μια πιο ρεαλιστική εμφάνιση. Η διαδικασία αυτή απαιτεί δειγματοληψία του χάρτη κανονικού διανύσματος για καθορισμό του normal, προβολή του διανύσματος προβολής (viewing vector) πάνω στο κανονικό διανυσμα, και χρήση του αποτελεσμάτος για offset στις συντεταγμένες uv.



Σχήμα 4.2: Η λογική του parallax mapping.

### 4.3 Weighted Voronoi Stippling

Weighted Voronoi Stippling είναι μια τεχνική παραγωγής μοτίβων stippling (ζωγραφισμένων με τελείες), που καθορίζεται από την απόσταση από ένα σύνολο σημείων ελέγχου. Κάθε control point έχει το δικό του βάρους, που επηρεάζει το μέγεθος και τη πυκνότητα των τελείων.



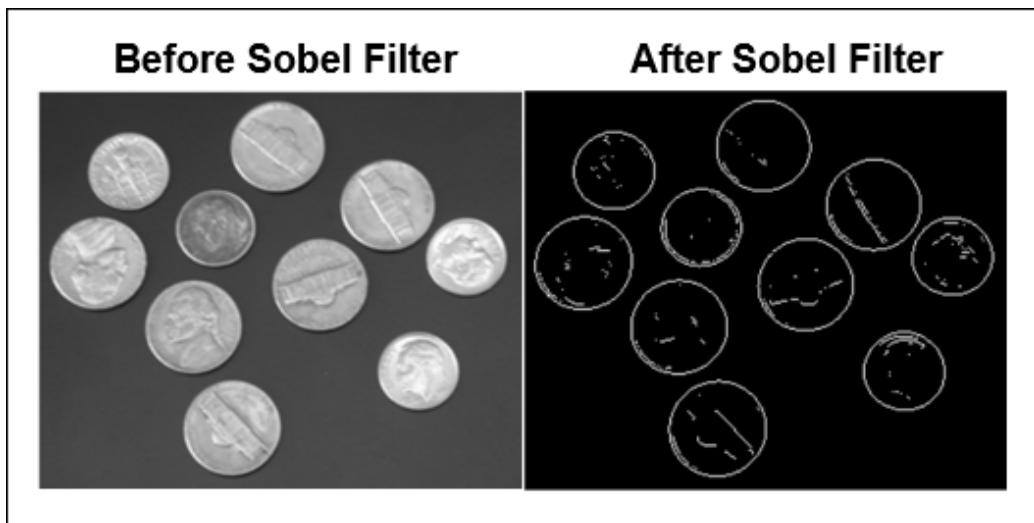
(a) Voronoi diagram generated by the set of generators (large dots). Centroids of each Voronoi region are marked by the small dots.

(b) Centroidal Voronoi diagram

Σχήμα 4.3: Μη σταθμισμένο διάγραμμα Voronoi, και η μετακίνηση των σημείων πρός το κεντροειδές.

## 4.4 Sobel filters

Τα φίλτρα Sobel χρησιμοποιούνται για edge detection μιας εικόνας, υπολογίζοντας το gradient της έντασης της εικόνας σε κάθε πίξελ. Χρησιμοποιούν παράθυρα (kernels) μεγέθους  $3 \times 3$  για εντοπισμό οριζόντιων και κάθετων ακμών, και το συνολικό αποτέλεσμα μιας δίνει όλες τις ακμές σε μια εικόνα - αν και είναι επιρρεπής στον θόρυβο.



Σχήμα 4.4: Εντοπισμός ακμών με χρήση Sobel filters.

# 5 Αποτελέσματα & Συζήτηση

## 5.1 Αποδόσεις

## 5.2 Αδυναμίες - Πιθανές Βελτιώσεις

Το πρόγραμμα απαιτεί περίπου 80 δευτερόλεπτα ώστε να φορτώσει, που οφείλεται στα υψηλής ανάλυσης textures που χρησιμοποιούνται, και στα μοντέλα πολλαπλών σημείων για τα δωμάτια.

Όλα τα δωμάτια, με την εξαίρεση του Studio Room και του Metallic Cave έχουν “baked in” την πληροφορία πέρι φωτισμού, πράγμα που τα καθιστά δύσχρηστα μαζί με τα μοντέλα φωτισμού μας (Phong).

Υπάρχει έλλειψη στο πρόγραμμά μας περίπλοκου φωτισμού και εφαρμογή σκιών, καθώς και επιπλέον αντικειμένων όπως ζητείται στο Ερώτημα 2.

Το εφέ Floyd-Steinberg Dithering και Pointilism, καθώς και σε λιγότερο ποσό το Kuwahara filter, είναι πάρα πολύ απαιτητικά στην εκτέλεση και οδηγούν πτώση στα frames. Επίλυση του προβλήματος αυτού είναι η βελτιστοποίηση του κώδικα και η αφαίρεση περιττών πράξεων.

Αν και ενδιαφέρον, το εφέ του πουαντιγισμού εφαρμοσμένο στο δωμάτιο Peirene Fountain είναι ημιτελειωμένο, και χρειάζεται εφαρμογή του Weighted Voronoi Stippling στην δειγματοληψία που πραγματοποιούμε.

Απαιτείται έλεγχος για την αποτυχία του εμφανούς αποτελέσματους του Parallax Mapping ή και διόρθωση.

## 6 Συμπεράσματα

Με την ολοκλήρωση της εκπόνησης της εργασίας, έχουμε μάθει να εργαζόμαστε με πολλαπλά μοντέλα και το απαιτούμενο texture mapping τους, και να τα προσανατολίζουμε στον χώρο. Έχουμε μάθει να χειρίζόμαστε επίσης πολλαπλούς framebuffers, να κατασκευάζουμε portals με τη τεχνική του attached texture σε framebuffer, να παράγουμε και να εφαρμόζουμε normal mapping και height maps και να ανυψώνουμε δισδιάστατες εικόνες σύμφωνα με αυτά.

Εξοικειωνόμαστε σε μεγάλο βαθμό με post-processing effects, με εκ των υστέρων τροποποίηση της παραγόμενης εικόνας, εκτενή χρήση φίλτρων και γενικότερα προγραμματισμό GLSL.

Τέλος, μαθαίνουμε να οργανώνουμε και να χειρίζόμαστε περίπλοκα και εκτενή προγράμματα OpenGL στην C++, καθώς και τη χρήση των CMake και του περιβάλλοντος Visual Studio.

# 7 Οδηγίες Χρήσης Προγράμματος

- Κατεβάζουμε και εγκαθιστούμε CMake και τον source code του προγράμματος.
- Κάνουμε build το VS solution του πηγαίου κώδικα σε έναν φάκελο build.
- Τρέχουμε το πρόγραμμα είτε μέσα από το Visual Studio ή εκτελούμε απευθείας το .exe αρχείο.

## 7.1 Για τη χρήση εντός του προγράμματος:

Instructions:

WASD: Move camera

Q&E: Move up and down

ENTER: Toggle active Dithering

SHIFT + ENTER: Toggle active colored Dithering or pre-dithered/original Texture

P: Toggle Pointilism

K: Toggle Kuwahara Filter

C: Toggle Comic Book Effect

T: Toggle CRT Effect

N: Toggle Normal Maps

M: Toggle Parallax Mapping

R: Raise Mesh

SHIFT + R: Stop Portal Cameras moving

O: Disable transitions and 1-7: Select transition effect

---

BACKSPACE: Print booleans status

SHIFT + BACKSPACE: Toggle minimap for debugging

LEFT/RIGHT ARROW: Change minimap texture

CTRL: Print Instructions Again

ESC: Quit

## 8 Βιβλιογραφία

- 3Δ Μοντέλα, <https://sketchfab.com>
- Moustakas, K., Paliokas, I., Tsakiris, A., & Tzovaras, D. (2015). Computer Graphics and Virtual Reality [Undergraduate textbook]. Kallipos, Open Academic Editions. <https://dx.doi.org/10.57713/kallipos-574>
- LearnOpenGL, <https://learnopengl.com>
- Toon Shader - Version II, <https://www.lighthouse3d.com/tutorials/glsl-12-tutorial/toon-shader-version-ii/>
- Charlotte Hoare (2012). Interactive non-photorealistic rendering using GLSL. Bournemouth University. <https://nccastaff.bournemouth.ac.uk/jmacey/MastersProject/MSc12/Hoare.pdf>
- Aaron Hertzmann. Stroke-Based Rendering. University of Washington. <https://www.dgp.toronto.edu/~hertzman/sbr02/hertzmann-sbr02.pdf>
- Itamar Turner-Trauring (2023). Speeding up your code when multiple cores aren't an option. PythonSpeed. <https://pythonspeed.com/articles/optimizing-dithering/>
- Pete's PSX GPU plugins. Pete's Domain. <http://www.pbernert.com/html/gpu.htm#SHADER>
- The Coding Train (2024). Coding Challenge 181: Weighted Voronoi Stippling. YouTube. [https://www.youtube.com/watch?v=Bxdt6T\\_1qgc](https://www.youtube.com/watch?v=Bxdt6T_1qgc)
- JEGX (2010). Geeks3D. <https://www.geeks3d.com/shader-library/>
- AgateDragon. Agate Dragon Games - Creating a halftone shader. <https://agatedragon.blog/2024/01/01/creating-a-halftone-shader/>
- AgateDragon. Shadertoy.  
Real-time oil painting with OpenGL, ROXLU, <https://www.roxlu.com/2013/012/real-time-oil-painting-with-opengl>

- Acerola (2022). This is the Kuwahara Filter. YouTube. <https://www.youtube.com/watch?v=LDrvJK3U9g>
- Babylon.js (2020). Retro CRT Shader — A post processing effect study. Medium. <https://babylonjs.medium.com/retro-crt-shader-a-post-processing-effect-study-1cb3f783afbc>
- Fabien Sanglard (2009). BumpMapping with GLSL. Fabien Sanglard's Website. <https://fabiensanglard.net/bumpMapping/>
- moonpatel2003. Normal Mapping in Computer Graphics. GeeksforGeeks. <https://www.geeksforgeeks.org/normal-mapping-in-computer-graphics/>
- GLTransitions, <https://gl-transitions.com>