ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΙΤΛΟΣ

“Προσομοίωση Έγχρωμου Τριγωνικού Matrix Barcode σε περιβάλλον Unity”

Παππάς Ευάγγελος (ΑΕΜ: 7003)

επιβλέπων καθηγητής: Ατρέας Νικόλαος

Θεσσαλονίκη 2023

Title in English:

“Simulation of a Colored Triangular Matrix Barcode in Unity”

Author: Pappas Evangelos

Supervisor: Nikolaos Atreas

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω θερμά τον καθηγητή μου Νικόλαο Ατρέα για την καθοδήγηση, τη συνέπεια και την εμπιστοσύνη που μου έδειξε. Η εκπόνηση της διπλωματικής μου εργασίας ήταν πιο ευχάριστη από τις πιο υψηλές μου προσδοκίες χάρη στη δική του παρουσία.

Θέλω επίσης να ευχαριστήσω την οικογένειά μου και τους φίλους μου και ιδιαίτερα τον Ισίδωρο Τσούλκα και Αριστοτέλη Μαυρόπουλο για τη συμπαράστασή τους όλα αυτά τα χρόνια.

Τέλος θα ήθελα να ευχαριστήσω όλους αυτούς που συνέβαλαν τα μέγιστα για να φτάσω ως εδώ και δεν είναι πια μαζί μας.

Από τα βάθη της καρδιάς μου, σας ευχαριστώ.

Περιεχόμενα

Abstract

Matrix Barcodes (and barcodes in general) have been a part of our everyday life for almost half a century. The QR Code (Quick Response Code) had a great rise in popularity thanks to the evolution of the processing power and camera capabilities of smartphones. The aim of this thesis is to simulate a novel Triangular Colored Matrix Barcode to examine the prospects of non square shapes in Matrix Barcodes as well as the benefits of using color to condense information in a smaller area. The document is divided in six chapters. In the first chapter, there is a presentation of principles and terminology which will be used throughout the document. It also contains a short description of some of the most influential Matrix Barcode that were an inspiration for this thesis. The second chapter includes a description of the encoding process at both theoretical and practical levels. This chapter is followed by a short description of the masking and scoring system that is being used in this particular Matrix Barcode. The fourth chapter describes and analyzes the decoding process which is the reverse procedure of encoding. The fifth chapter presents a simulation of a more real life situation where the Matrix Barcode is decoded using a screenshot. The sixth and final chapter of this thesis presents conclusions, problems and possible improvements of this Matrix Barcode.

Περίληψη

[1][2]

Τα Matrix Barcode, ή αλλιώς οι κώδικες δύο διαστάσεων, αποτελούν μια εξέλιξη των κλασικών μονοδιάστατων barcode (γραμμωτός κώδικας) και είναι μέθοδοι απεικόνισης δεδομένων σε μορφή που είναι αναγνώσιμη από μηχανή.

Η πρώτη επιτυχής εγκατάσταση ενός συστήματος με barcode έγινε σε έναν όμιλο υπεραγορών στη Μεγάλη Βρετανία το 1972. Η ταχύτητα ανάγνωσης δεδομένων από τη μηχανή καθώς και η ευκολία αυτής της διαδικασίας, αύξησε τη δημοτικότητα των Barcode καθώς εξοικονομούσε χρήματα στις επιχειρήσεις. Το 1994 μια ομάδα επιστημόνων στην Ιαπωνική εταιρία Denso Wave εφηύρε το QR Code (Quick Response Code), υπό την επίβλεψη του Masahiro Hara. Το QR (που σημαίνει κώδικας ταχείας απόκρισης) είναι ένα ασπρόμαυρο Matrix Barcode 2D (δισδιάστατο). Η άνοδος της δημοτικότητας των έξυπνων τηλεφώνων (Smartphone) που ακολούθησε και άρα η εύκολη πρόσβαση σε φωτογραφική κάμερα και υπολογιστική ισχύ, επέφερε και ραγδαία αύξηση της δημοτικότητας του QR Code. Το αποκορύφωμα αυτής της αύξησης ήρθε με την πανδημία του ιού COVID-19 όπου τα πιστοποιητικά εμβολιασμού και νόσησης αποτυπώνονταν με τη βοήθεια του QR Code.

Την τελευταία δεκαετία έχουν καθιερωθεί στην καθημερινότητα και, παράλληλα, έχουν εφευρεθεί παραλλαγές του QR Code αλλά και των Matrix Barcode γενικότερα για να εξυπηρετούν καλύτερα τον εκάστοτε σκοπό τους. Η εξέλιξη των δυνατοτήτων των smartphone -τόσο σε υπολογιστική ισχύ όσο και στις προδιαγραφές της κάμερας- καθιστά πλέον δυνατή την εφεύρεση πιο πολύπλοκων Matrix Barcode που ξεφεύγουν από την κλασική προσέγγιση ασπρόμαυρων εικόνων και απεικονίσεων bit (δυαδικών ψηφίων).

Ο στόχος λοιπόν της παρούσης διπλωματικής είναι η ανάπτυξη ενός τριγωνικού και έγχρωμου γραμμωτού κώδικα δύο διαστάσεων (Triangular Colored Matrix Barcode) για την κωδικοποίηση μηνυμάτων γραμμένων στο αλφάβητο χαρακτήρων ASCII 7-bit. Η προσομοίωση της κωδικοποίησης και αποκωδικοποίησης πραγματοποιείται στο περιβάλλον του Unity 3D Game Engine για το λειτουργικό σύστημα των Windows. Ο χρήστης πληκτρολογεί μια φράση στο πεδίο κειμένου της διεπαφής και αφού η φράση ελεγχθεί για την εγκυρότητά της, κωδικοποιείται σύμφωνα με τις υπάρχουσες προδιαγραφές και αποκωδικοποιείται πατώντας ένα άλλο κουμπί στη διεπαφή χρήστη. Για τις ανάγκες της προσομοίωσης του “σκαναρίσματος”, το πρόγραμμα παρέχει τη δυνατότητα αποκωδικοποίησης από την επεξεργασία εικόνας του Matrix Barcode που παρήχθη μέσω της εξαγωγής στιγμιότυπου εικόνας (screenshot) σε ιδανικές συνθήκες.

Η επιλογή της προσθήκης χρωμάτων στο Matrix Barcode έγινε για την αύξηση της αναλογίας της πληροφορίας προς τον χώρο. Ένα στοιχείο ενός ασπρόμαυρου Matrix Barcode μπορεί να πάρει, άρα να αποθηκεύσει, μόνο δύο τιμές (άσπρο ή μαύρο). Ένα στοιχείο ενός έγχρωμου Matrix Barcode μπορεί να αποθηκεύσει τόσες τιμές, όσο και τα διαθέσιμα έγκυρα χρώματα που ορίζονται στις προδιαγραφές του. Η επιλογή του ισοσκελούς τριγώνου έγινε για την υποβοήθηση της εύρεσης του ορθού προσανατολισμού απ’ όπου αρχίζουμε την κωδικοποίηση και

αποκωδικοποίηση, όπως θα αναλυθεί στη συνέχεια του εγγράφου. Πέραν της προφανούς αισθητικές (οπτικής) αναβάθμισης του κώδικα, σε σχέση με τις κλασικές και πιο αυστηρές μορφές των Matrix Barcode, η κατασκευή αυτή πιστεύουμε να είναι χρήσιμη για εμπορικούς σκοπούς.

Η παρούσα διπλωματική χωρίζεται σε έξι κεφάλαια.

Στο πρώτο κεφάλαιο παρουσιάζουμε βασικές εισαγωγικές έννοιες οι οποίες θα χρησιμοποιούνται σε όλη την έκταση του εγγράφου. Επιπρόσθετα, το κεφάλαιο αυτό περιλαμβάνει μια σύντομη ανάλυση των πιο σημαντικών Matrix Barcode που υπάρχουν, τα οποία αποτέλεσαν και την έμπνευση για το Matrix Barcode που δημιουργήθηκε.

Στο δεύτερο κεφάλαιο ασχολούμαστε με το θεωρητικό κομμάτι της κωδικοποίησης και τη μεταφορά του στο περιβάλλον του Unity. Ακόμη, αναλύουμε την επιλογή των χρωμάτων και την ανάθεση των τιμών σε κάθε χρώμα. Στο κεφάλαιο παρουσιάζονται, επίσης, σε μορφή ψευδογλώσσας οι πιο σημαντικές προγραμματιστικές κλάσεις που είναι μέρη της κωδικοποίησης και είναι γραμμένα - στην πρωτότυπή τους μορφή - σε γλώσσα προγραμματισμού C# (C Sharp).

Στο τρίτο κεφάλαιο ασχολούμαστε με το Masking (απόκρυψη δεδομένων) και τη χρησιμότητά του στο παρόν Matrix Barcode. Παρουσιάζουμε τέσσερις μάσκες, τους κανόνες και το σύστημα βαθμολογίας τους και αναλύουμε τα αποτελέσματά τους.

Στο τέταρτο κεφάλαιο πραγματοποιείται η περιγραφή και η ανάλυση της αποκωδικοποίησης τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο.

Το πέμπτο κεφάλαιο εμπεριέχει μια προσομοίωση της διαδικασίας του “σκαναρίσματος” που περιλαμβάνει την περιγραφή της εξαγωγής του στιγμιότυπου εικόνας και της επεξεργασίας αυτού, ώστε να είναι εφικτή η αποκωδικοποίησή του.

Τέλος, το έκτο κεφάλαιο περιέχει παρατηρήσεις, συμπεράσματα, πιθανά προβλήματα καθώς και μελλοντικές βελτιώσεις που μπορούν να γίνουν στο παρόν Matrix Barcode.

Κεφάλαιο 1. Εισαγωγή

Στο κεφάλαιο αυτό, και συγκεκριμένα στη παράγραφο 1.1, γίνεται μια αναδρομή και σύντομη ανάλυση των πιο σημαντικών Matrix Barcode που υπάρχουν και επηρέασαν την κατασκευή του Matrix Barcode που προτείνουμε στη συνέχεια. Στην παράγραφο 1.2, εισάγουμε βασικές έννοιες και χαρακτηριστικά τα οποία είναι απαραίτητα για την κατανόηση των Matrix Barcode. Τα χαρακτηριστικά αυτά είναι επίσης αναγκαία για την αξιόπιστη λειτουργία των Matrix Barcode και υποβοηθούν την ανάγνωση του κώδικα από μια συσκευή.

1.1 Έγχρωμα υπάρχοντα Matrix Barcode

Υπάρχουν Matrix Barcode τα οποία αποτέλεσαν έμπνευση για τη δημιουργία του Matrix Barcode της παρούσης διπλωματικής. Ακολουθεί μια σύντομη περιγραφή των πιο σημαντικών από αυτά:

1.1.1 QR code

[2] Το QR code αποτελεί σίγουρα το πιο δημοφιλές Matrix Barcode. Εφευρέθηκε το 1994 από την Ιαπωνική εταιρία Denso Wave για να χρησιμοποιηθεί ως μέσο αναγνώρισης “ετικέτα” για εξαρτήματα αυτοκινήτων, αλλά σταδιακά άρχισε να επεκτείνεται η χρήση του με αποκορύφωμα την πανδημία του ιού COVID το 2019.

Το QR code αποτελείται από μαύρα τετράγωνα σε λευκό φόντο τοποθετημένα σε ένα τετραγωνικό πλέγμα τα οποία αναπαριστούν δεδομένα (data), ενώ κάποια από αυτά χρησιμοποιούνται ως λειτουργικά στοιχεία για τη ρύθμιση σημαντικών παραμέτρων κατά την αποκωδικοποίηση του κώδικα.

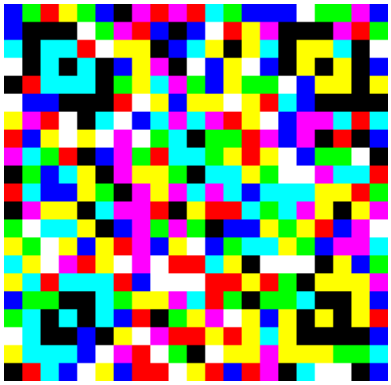
Χρησιμοποιώντας μια συσκευή απεικόνισης (όπως μια κάμερα κινητού), και αφού γίνει επεξεργασία μέσω ενός αλγορίθμου διόρθωσης σφάλματος Reed-Solomon, η αποκωδικοποίηση γίνεται γρήγορα και αξιόπιστα.

Σχήμα 1: *QR code για το URL (διεύθυνση ιστοσελίδας) της αγγλόφωνης Wikipedia σε έκδοση κινητού*

1.1.2 JAB code

[3] [4] Το JAB code είναι ένα από τα λίγα έγχρωμα Matrix Barcodes που υπάρχουν. Αποτελείται από τετράγωνα στοιχεία τοποθετημένα σε τετραγωνικό ή ορθογωνικό πλέγμα και αναπτύχθηκε στο Fraunhofer Institute SIT (Secure Information Technology) από το 2019 μέχρι το 2022. Ο κώδικας χρησιμοποιεί 4 ή 8 χρώματα του CMYK (Cyan, Magenta, Yellow, Key) χρωματικού μοντέλου (κυανό, ματζέντα, κίτρινο και μαύρο). Τα υπόλοιπα 4 χρώματα (κόκκινο, μπλε, πράσινο και λευκό) αποτελούν τα δευτερεύοντα χρώματα του CMYK μοντέλου. Ως αλγόριθμος διόρθωσης σφαλμάτων για το JAB code, επιλέχθηκε ο Low Density Parity Check (LDPC) που κρίθηκε καταλληλότερος για έγχρωμα Matrix Barcode.

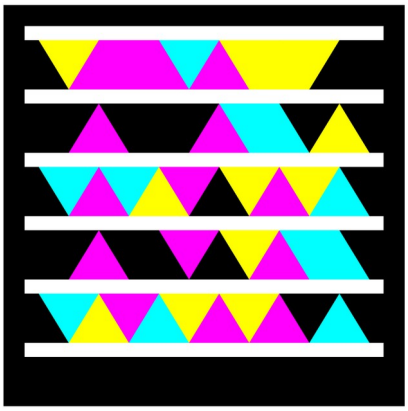
Σε σύγκριση με άλλα συμβατικά ασπρόμαυρα Matrix Barcodes, το JAB code έχει τη δυνατότητα να περιέχει στην ίδια περιοχή του κώδικα δύο ή τρεις φορές περισσότερες πληροφορίες για χρήση τεσσάρων ή οχτώ χρωμάτων αντίστοιχα. Με βάση αυτή τη δυνατότητά του, είναι ακόμα πιο εφικτή η αποθήκευση ενός ολόκληρου μηνύματος στον κώδικα και όχι μόνο μια αναφορά στο μήνυμα αυτό, όπως π.χ. ένα σύνδεσμο για μία ιστοσελίδα.



Σχήμα 2: *Χαιρετισμός της Wikipedia κωδικοποιημένος με JAB code 8 χρωμάτων*

1.1.3 High Capacity Color Barcode

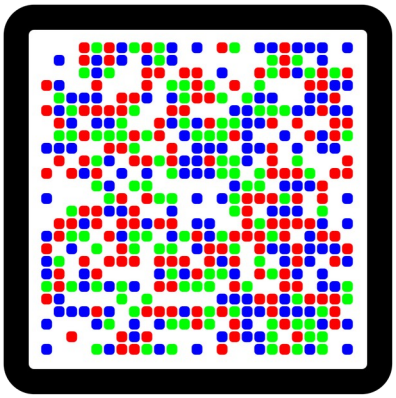
[5] Το High Capacity Color Barcode αποτελεί μια τεχνολογία που αναπτύχθηκε από τη Microsoft για την κωδικοποίηση δεδομένων χρησιμοποιώντας συστάδες έγχρωμων τριγώνων αντί των συμβατικών τετραγώνων. Η πυκνότητα πληροφορίας είναι μεγαλύτερη σε σύγκριση με τα μονόχρωμα Matrix Barcodes καθώς χρησιμοποιείται μια παλέτα με 2, 4 ή 8 χρώματα. Δημιουργήθηκε το 2007 από τον Gavin Jancke και -όπως δήλωσε ο ίδιος- δεν είχε σκοπό να αντικαταστήσει τα συμβατικά barcodes αλλά να προσφέρει κάτι πιο εξειδικευμένο στην εταιρία.



Σχήμα 3: *Ένα παράδειγμα ενός High Capacity Color Barcode*

1.1.4 Cronto Visual Cryptogram

[6] Το Cronto Visual Cryptogram (ή αλλιώς PhotoTAN) είναι ένα εξειδικευμένο έγχρωμο barcode που αποτελεί προϊόν έρευνας του πανεπιστημίου του Cambridge. Αποτελείται από έγχρωμες βούλες τεσσάρων χρωμάτων (κόκκινο, πράσινο, μπλε και λευκό) οι οποίες είναι τοποθετημένες σε ένα τετραγωνικό πλέγμα. Χρησιμοποιείται για συναλλαγές σε e-banking και προορίζεται μόνο για ψηφιακή και όχι εκτυπώσιμη μορφή. Εφαρμόζοντας μεθόδους κρυπτογράφησης, εγγυάται την ασφάλεια και την εμπιστευτικότητα τραπεζικών συναλλαγών. Ταυτόχρονα βελτιώνει την εμπειρία του χρήστη, καθώς απλοποιεί τις τραπεζικές διαδικασίες.



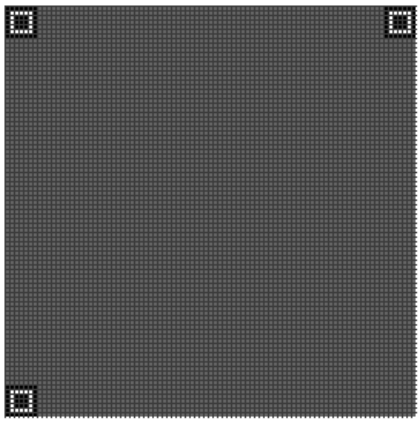
Σχήμα 4: *Μία απεικόνιση ενός photoTAN code με τυχαία δεδομένα*

1.2 Βασικά χαρακτηριστικά των Matrix Barcode

[7] Τα προαναφερθέντα αλλά και άλλα δισδιάστατα Barcode, έχουν κάποια βασικά χαρακτηριστικά. Τα χαρακτηριστικά αυτά εξυπηρετούν τις ανάγκες του κάθε τύπου Matrix Barcode και είναι απαραίτητα για την αξιόπιστη ανάγνωσή τους. Αυτό σημαίνει ότι υπάρχουν Matrix Barcode τα οποία διαθέτουν όλα αυτά τα χαρακτηριστικά ενώ κάποια άλλα έχουν λιγότερα λόγω του τρόπου χρήσης τους. Ακολουθεί μια σύντομη περιγραφή των πιο σημαντικών χαρακτηριστικών.

1.2.1 Finder Pattern (μοτίβο εύρεσης)

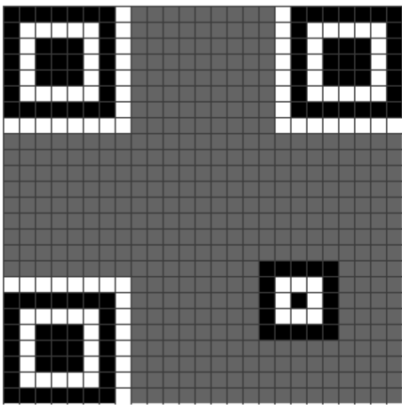
Το Finder Pattern ή αλλιώς μοτίβο εύρεσης αποτελεί ίσως το πιο βασικό χαρακτηριστικό των Matrix Barcode. Ανήκει στην κατηγορία των function patterns (λειτουργικά μοτίβα) που είναι ουσιαστικά συγκεκριμένα σχήματα τα οποία εμφανίζονται σε προκαθορισμένα μέρη του κώδικα που δεν περιέχουν δεδομένα. Το Finder Pattern χρησιμεύει στον εντοπισμό του κώδικα και της θέσης αυτού από μια συσκευή απεικόνισης (π.χ. μια κάμερα) και συχνά βοηθούν και στην ανίχνευση του ορθού προσανατολισμού του. Τα μοτίβα εύρεσης επιλέγονται από τον κατασκευαστή με τέτοιο τρόπο ώστε να μην εμφανίζονται συχνά ως περιεχόμενα του κώδικα αυτού. Κατά συνέπεια τα μοτίβα σπανίως θα εμφανιστούν τυχαία σε σημεία που αποθηκεύεται η πληροφορία ώστε να μη γίνεται λανθασμένη ανάγνωση από τη μηχανή.



Σχήμα 5: *Τα τρία Finder Pattern στις γωνίες ενός QR Code Version 18*

1.2.2 Alignment Pattern (μοτίβο ευθυγράμμισης)

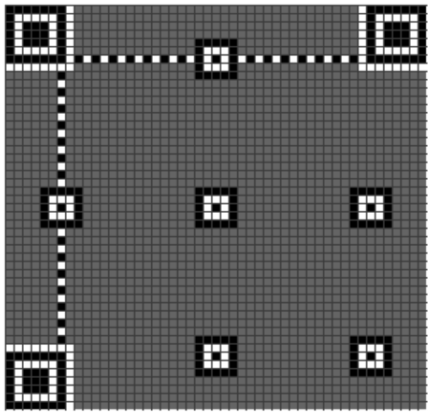
Το Alignment pattern ή μοτίβο ευθυγράμμισης είναι ένα ακόμη βασικό χαρακτηριστικό των Matrix Barcodes. Ανήκει κι αυτό στην κατηγορία των function patterns και η λειτουργία του είναι να βοηθάει στην ανίχνευση του ορθού προσανατολισμού από μια συσκευή απεικόνισης. Ουσιαστικά, η λειτουργία του είναι να ευθυγραμμίζει τα μέρη της πληροφορίας με τα Finder Patterns. Συνήθως, όσο πιο μεγάλο σε διαστάσεις είναι το Matrix Barcode, τόσο περισσότερα alignment pattern περιέχει στα σημεία που έχει προνοήσει ο κατασκευαστής.



Σχήμα 6: *Ένα Alignment Pattern μαζί με τρία Finder Pattern ενός QR Code*

1.2.3 Timing Pattern (μοτίβο χρονισμού)

Το Timing pattern ή μοτίβο χρονισμού αποτελεί την εφαρμογή ενός μηχανισμού διαχωρισμού των σειρών και στηλών ενός Matrix Barcode. Γενικά χρησιμοποιούνται στον κατακόρυφο και οριζόντιο άξονα για την ευκολότερη αντιστοίχιση στοιχείων δεδομένων σε συντεταγμένες. Το timing pattern συναντάται σε μερικά Matrix Barcode και ως clock pattern.



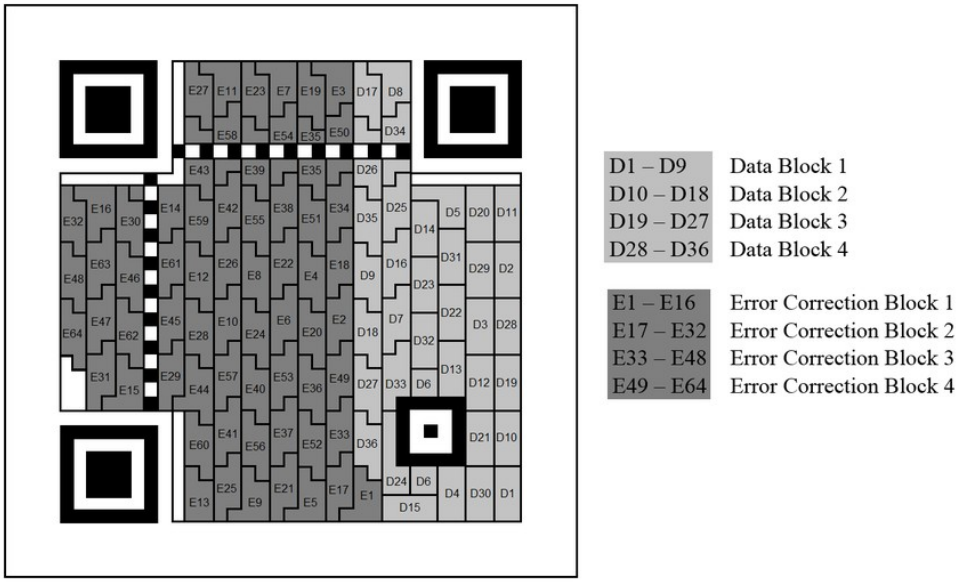
Σχήμα 7: *Timing patterns στον οριζόντιο και κατακόρυφο άξονα μαζί με τα τρία Finder Pattern και έξι alignment pattern ενός QR Code*

1.2.4 Error Correction Code (κώδικας διόρθωσης σφάλματος)

Το Error Correction Code ή κώδικας διόρθωσης σφάλματος είναι ένας κώδικας ο οποίος εφαρμόζει μια τεχνική διόρθωσης σφάλματος. Αποτελεί ένα από τα βασικότερα και πιο αναγκαία χαρακτηριστικά ενός Matrix Barcode. Οι τεχνικές διόρθωσης σφάλματος αποτελούν έναν μεγάλο

επιστημονικό κλάδο και είναι αναγκαίες σε κάθε κανάλι επικοινωνίας που εμπεριέχεται θόρυβος. Ίσως από τους πιο διαδεδομένους κώδικες διόρθωσης σφάλματος είναι ο Solomon-Reed ο οποίος χρησιμοποιείται στα CD (Compact Discs) και στο QR Code και είναι κατάλληλος για τη διόρθωση σειριακών σφαλμάτων (burst errors). Στο JAB Code χρησιμοποιείται ο LDPC (Low Density Parity Check) ο οποίος εντοπίζει και διορθώνει κυρίως τυχαία σφάλματα (random errors).

Οι περισσότεροι κώδικες διόρθωσης σφάλματος χρησιμοποιούν parity bits (bit ισοτιμίας) τα οποία προκύπτουν μετά από επεξεργασία των δεδομένων. Αξιοποιώντας αυτή την επιπρόσθετη πληροφορία, ο αποκωδικοποιητής μπορεί να ανιχνεύσει και να διορθώσει τυχόν σφάλματα που προέκυψαν κατά τη διάρκεια της αποκωδικοποίησης. Σε πολλές περιπτώσεις, ο χρήστης έχει την επιλογή να διαλέξει το επίπεδο του Error Correction (χαμηλό, μεσαίο, υψηλό), γεγονός το οποίο κάνει πιο αξιόπιστο έναν κώδικα (σε περίπτωση υψηλού επιπέδου error correction) ή πιο μικρό από άποψη χώρου αντίστροφα. Ενδεικτικά, στο παρακάτω QR Code, τα Error Correction block σημειώνονται με την ακολουθία E1-E64 ενώ τα block δεδομένων με την ακολουθία D1-D36.



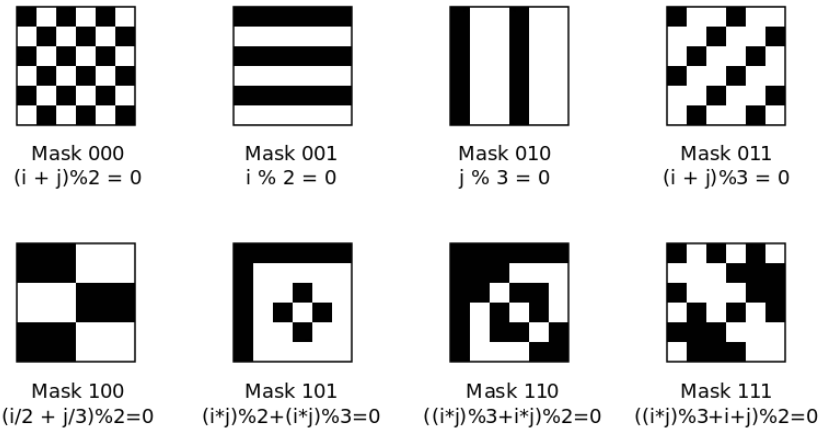
Σχήμα 8: *Ένα QR Code με τα "block" δεδομένων του (σημειωμένα με D) και τα "block" του error correction (σημειωμένα με E)*

1.2.5 Masking

Το Masking είναι μια τεχνική επικάλυψης και απόκρυψης της πληροφορίας. Χρησιμοποιείται σε πολλά Matrix Barcode για την αποφυγή εμφάνισης ιδίων ή παρόμοιων κωδικοποιημένων μοτίβων που μπορεί να οδηγήσουν σε πρόβλημα σωστής αναγνώρισης, άρα και αποκωδικοποίησης. Για παράδειγμα, σε ένα QR Code η πληροφορία μπορεί να πάρει την τιμή άσπρο ή μαύρο. Εφαρμόζοντας μια μάσκα στην πληροφορία, είναι εφικτή η αποφυγή συνεχόμενων ίδιων τιμών που θα κάνει πιο εύκολα διαχωρίσιμα τα bits μεταξύ τους και άρα πιο αξιόπιστη την αποκωδικοποίησή τους. Χρησιμοποιώντας μια τεχνική μάσκας (masking), η οποία αλλάζει τη μορφή εμφάνισης για τέτοιοι τύπου δυσλειτουργίες ακολουθώντας συγκεκριμένους κανόνες,

δημιουργούμε κώδικα ο οποίος είναι πιο στέρεος (stable) σε σφάλματα αποκωδικοποίησης και αποκρύπτει την πληροφορία.

Στη μεγαλύτερη πλειοψηφία των Matrix Barcode, υπάρχουν ενσωματωμένα στο πρόγραμμα κανάλια (presets) masking. Μετά την αρχική κωδικοποίηση, γίνεται αξιολόγηση του κάθε καναλιού masking και χρησιμοποιείται το βέλτιστο (με κάποια έννοια) κανάλι για την εκάστοτε περίπτωση. Αφού επιλεχθεί ένα συγκεκριμένο κανάλι, τυπώνεται στον κώδικα με κάποιο κατάλληλο τρόπο έτσι ώστε να είναι δυνατή η αναγνώριση της μάσκας, για να μπορεί να γίνει σωστά η αντίστροφη διαδικασία από τον αποκωδικοποιητή.



Σχήμα 9: *Οχτώ διαφορετικές μάσκες του QR Code. Κάθε μάσκα αντιστοιχεί σε μια ακολουθία τριών μπιτ και έχει μια συνάρτηση που την ορίζει. Στα παραπάνω σχήματα, συμβολίζουμε με i και j τις γραμμές και στήλες πίνακα αντιστοίχως και με % την πράξη Modulo.*

Κεφάλαιο 2. Κωδικοποίηση

Στο κεφάλαιο της κωδικοποίησης ξεκινάμε με την κωδικοποίηση χαρακτήρα στην παράγραφο 2.1. Δικαιολογούμε την επιλογή της κωδικοποίησης ASCII 7-bit και τη συνδέουμε με την κωδικοποίηση σε Matrix Barcode στην παράγραφο 2.2. Στη συνέχεια παρουσιάζουμε στην παράγραφο 2.3 το κλασικό μοντέλο RGB απεικόνισης χρωμάτων και αναλύουμε το 7-bit RGB μοντέλο που χρησιμοποιούμε. Ακολουθεί ο υπολογισμός των τιμών των χρωματικών αποχρώσεων του κόκκινου, πράσινου και μπλε για το Matrix Barcode που κατασκευάζουμε στα πλαίσια του τρόπου αναπαράστασης των χρωμάτων στο Unity δημιουργώντας τον Πίνακα 1 της παραγράφου 2.4. Η παράγραφος 2.5 περιλαμβάνει τη διαδικασία κωδικοποίησης του χαρακτήρα στο Unity ενώ η επόμενη παράγραφος περιγράφει τα κριτήρια εγκυρότητας ενός μηνύματος προς κωδικοποίηση. Στην παράγραφο 2.7 ολοκληρώνεται η κωδικοποίηση χωρίς τη διαδικασία του masking και στην τελευταία παράγραφο του κεφαλαίου παρουσιάζουμε το περιβάλλον χρήστη της εφαρμογής και το αποτέλεσμα της κωδικοποίησης ενός μηνύματος.

2.1 Κωδικοποίηση χαρακτήρα (Character Encoding)

[8]Η κωδικοποίηση, σαν όρος, είναι η διαδικασία μετατροπής πληροφοριών σε μια διαφορετική μορφή η οποία είναι κατάλληλη για την αποθήκευση ή τη μετάδοσή τους. Ακολουθεί ένα σύνολο κανόνων που καθιστά εφικτή την αντίστροφη διαδικασία (αυτή της αποκωδικοποίησης) και ταυτόχρονα μπορεί να αποκρύψει (αν αυτό είναι επιθυμητό) τη μεταδιδόμενη πληροφορία.

Η κωδικοποίηση χαρακτήρων είναι μια διαδικασία ανάθεσης αριθμών σε μια ομάδα γραφικών χαρακτήρων. Οι χαρακτήρες αυτοί συνήθως αποτελούν μέρος του αλφαβήτου κάποιας ανθρώπινης γλώσσας ή αριθμοί ή ακόμα και χαρακτήρες ελέγχου (control characters) οι οποίοι δεν είναι εκτυπώσιμοι και χρησιμοποιούνται για την μετάδοση εντολών και γεγονότων.

Για το Matrix Barcode που αναπτύσσουμε, επιλέγουμε την κωδικοποίηση χαρακτήρων ASCII 7-bit. Η κωδικοποίηση American Standard Code for Information Interchange, ή αλλιώς ASCII, εκδόθηκε το 1963 στην Αμερική και συνέχισε να αναπτύσσεται ως και το 1986. Η 7-bit έκδοσή της αποτελεί ουσιαστικά την κωδικοποίηση 128 = 2^7 χαρακτήρων οι οποίοι συμπεριλαμβάνουν όλο το αγγλικό αλφάβητο καθώς και σημεία στίξης, αριθμούς και άλλους ειδικούς χαρακτήρες. Οι 32 πρώτοι χαρακτήρες είναι χαρακτήρες ελέγχου οι οποίοι δεν είναι εκτυπώσιμοι.

Αυτή η έκδοση χρησιμοποιείται ως βάση για πολλές διαφορετικές κωδικοποιήσεις περισσοτέρων bit και είναι η κωδικοποίηση που χρησιμοποιείται στην αποστολή των URL (Uniform Resource Locator) κατά την πλοήγησή μας στο διαδίκτυο χρησιμοποιώντας έναν περιηγητή ιστού (browser). Η ASCII 7-bit κωδικοποίηση αποτελεί λοιπόν τη βάση η οποία θα χρησιμοποιηθεί για την ανάπτυξη του Matrix Barcode καθώς είναι ευρέως διαδεδομένη και παρουσιάζει πολλές ομοιότητες με τις περισσότερες κωδικοποιήσεις χαρακτήρων.

DEC	OCT	HEX	BIN	Symbol
32	040	20	00100000	␣
33	041	21	00100001	!
34	042	22	00100010	"
35	043	23	00100011	#
36	044	24	00100100	$

Σχήμα 10: *Δείγμα του πίνακα ASCII 7-bit*

2.2 Κωδικοποίηση σε Matrix Barcode

[1] Η κωδικοποίηση σε Matrix Barcode αφορά στη μετατροπή δεδομένων από τη binary (δυαδική) μορφή τους με την οποία αποθηκεύονται σε έναν υπολογιστή σε μια οπτικά αναγνωρίσιμη από μηχανή εικόνα. Η διαδικασία αυτή προφανώς υπόκειται σε κανόνες ο κυριότερος εκ των οποίων είναι η 1-1 αντιστοιχία για να είναι εφικτή και η αποκωδικοποίηση, όπως και η διασφάλιση του υπολογισμού του τύπου αντιστροφής. Προφανώς, η κωδικοποίηση σε Matrix Barcode παρέχει τη δυνατότητα αποθήκευσης περισσότερων πληροφοριών, σε σχέση με το παραδοσιακό μονοδιάστατο Barcode, καθώς αξιοποιείται μία ακόμα διάσταση για την αποτύπωση δεδομένων.

Έχοντας, λοιπόν, ως δεδομένα το σύνολο των ASCII 7-bit χαρακτήρων, σε κάθε αριθμημένο χαρακτήρα αντιστοιχούμε τη δυαδική μορφή του. Όπως αναφέραμε προηγουμένως, τα συνήθη Matrix Barcodes χρησιμοποιούν μόνο το άσπρο και το μαύρο χρώμα για την απεικόνιση των πληροφοριών για τις δύο πιθανές τιμές που μπορεί να πάρει ένα bit. Στην περίπτωση του Matrix Barcode που αναπτύσσουμε, η προσθήκη χρωμάτων αυξάνει 64 φορές την αναλογία πληροφορίας προς το χώρο που χρησιμοποιείται.

2.3 Μοντέλο RGB απεικόνισης χρωμάτων

Κάθε ASCII 7-bit χαρακτήρας αντιστοιχεί σε ένα μοναδικό φυσικό αριθμό από το 0 έως και το 127. Αυτό είναι φυσική συνέπεια της ύπαρξης 7 bit καθώς το νούμερο 1111111 στο δυαδικό σύστημα περιγράφει τον αριθμό 127 στο δεκαδικό σύστημα.

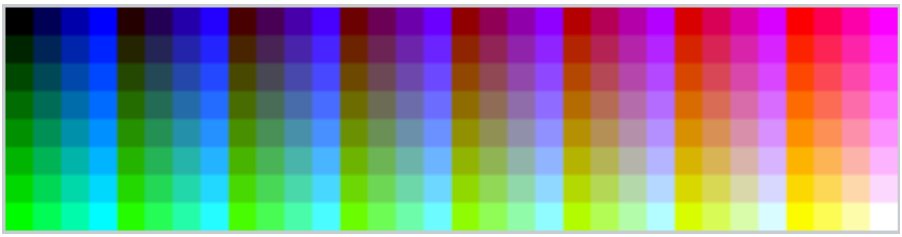
[9] Όπως είναι γνωστό, ένα από τα πιο διαδεδομένα μοντέλα περιγραφής χρωμάτων είναι το RGB (Red, Green, Blue) σύμφωνα με το οποίο είναι εφικτή η δημιουργία οποιασδήποτε απόχρωσης στο ανθρώπινο ορατό φάσμα. Το μοντέλο αυτό χρησιμοποιεί τα χρώματα κόκκινο, πράσινο και μπλε για την περιγραφή ενός χρώματος. Οι αποχρώσεις κάθε χρώματος μπορούν να πάρουν τιμές από μηδέν μέχρι και ένα μέγιστο το οποίο ορίζεται από το εκάστοτε λογισμικό που χρησιμοποιείται. Στην περίπτωση του Unity, ο τρόπος αναπαράστασης του χρώματος είναι το RGBA όπου το A (Alpha) αφορά την αδιαφάνεια ενός χρώματος που δε θα μας απασχολήσει καθώς θέλουμε όλα τα χρώματα να διακρίνονται καθαρά και η τιμή του A θα είναι πάντα η μέγιστη. Η τιμή κάθε χρώματος αναπαρίσταται με έναν δεκαδικό αριθμό (float) ο οποίος έχει εύρος τιμών από το 0 μέχρι και το 1. Έτσι για παράδειγμα, παραλείποντας το A, το άσπρο χρώμα θα έχει την τιμή (1, 1, 1), το μαύρο χρώμα την τιμή (0, 0, 0) και το κόκκινο χρώμα θα έχει την τιμή (1, 0 , 0).

Εφόσον τα χρώματα αναπαρίστανται με “συνεχή” τρόπο, χρειάζεται να γίνει μια ανάθεση χρώματος σε κάθε χαρακτήρα ASCII με διακριτό τρόπο.

[10] Μία από τις πιο βασικές μορφές του RGB είναι το 8-bit RGB ή αλλιώς 256 (2^8) χρώματα. Σύμφωνα με το μοντέλο αυτό χρησιμοποιούνται 3 bit για την αναπαράσταση του κόκκινου χρώματος, 3 bit για την αναπαράσταση του πράσινου χρώματος και 2 bit για την αναπαράσταση του μπλε χρώματος. Αυτό σημαίνει ότι το κόκκινο μπορεί να έχει 8 =2^3 διαφορετικές αποχρώσεις -όπως και το πράσινο- ενώ το μπλε μπορεί να έχει 4 = 2^2 αποχρώσεις.

Bit	7	6	5	4	3	2	1	0
Data	R	R	R	G	G	G	B	B

Σχήμα 11: *Τα bit αναπαράστασης κάθε χρώματος σε ένα 8-bit color μοντέλο. Με R σημειώνεται το κόκκινο, με G το πράσινο και με B το μπλε χρώμα.*



Σχήμα 12: *Το εύρος των χρωμάτων που μπορεί να απεικονίσει το 8-bit color μοντέλο.*

Έχοντας υπόψιν αυτό το μοντέλο και γνωρίζοντας ότι οι χαρακτήρες ASCII 7-bit είναι 128, επιλέχθηκε ένα μοντέλο χρωμάτων των 7 bit το οποίο σημαίνει ότι για κάθε χαρακτήρα ASCII 7-bit αντιστοιχεί ένα μόνο διακριτό χρώμα στο RGB. Εν τέλει, δημιουργείται ο παρακάτω πίνακας:

Bit	7	6	5	4	3	2	1	0
Data	-	R	R	G	G	G	B	B

(R = red, G = green, B = blue)

Σχήμα 13: *Ανάθεση bit σε χρώμα στο 7-bit color μοντέλο*

Εφόσον έχουμε διαθέσιμα 2 bit στο κόκκινο χρώμα, 3 bit στο πράσινο και 2 bit στο μπλε, μπορούμε να αναθέσουμε 2^2, 2^3 και 2^2 διακριτές τιμές έντασης κάθε χρώματος.

2.4 Υπολογισμός των τιμών χρώματος στο Unity

Το κάθε χρώμα περιγράφεται στο Unity με έναν float (δεκαδικό) αριθμό από 0 έως και 1. Χρησιμοποιώντας 2 bit για το μπλε μπορούμε να περιγράψουμε 4 διαφορετικές καταστάσεις για το χρώμα αυτό (00, 01, 10, 11) σε δυαδική μορφή. Επειδή χρειάζεται να χωρίσουμε το κλειστό διάστημα [0, 1] σε 4 μέρη, παίρνουμε (κατά προσέγγιση) τις float τιμές

$$\frac{0}{3}=0,\frac{1}{3}\approx 0.33,\frac{2}{3}\approx 0.67\,\kappa\alpha\iota\,\frac{3}{3}=1.$$

Οι ίδιες τιμές θα χρησιμοποιηθούν και για το κόκκινο, ενώ για το πράσινο οι 8 τιμές θα είναι (διαιρώντας με το 7) κατά προσέγγιση

0, 0.14, 0.28, 0.43, 0.57, 0.71, 0.86 και 1.

Έτσι, έχουμε τον πίνακα αποχρώσεων:

	Κ = Κόκκινο	**Π = Πράσινο**	**Μ =Μπλε**
0	**$K_0 = 0$**	**$П_0 = 0$**	**$M_0 = 0$**
1	**$K_1 = 0.33$**	**$П_1 = 0.14$**	**$M_1 = 0.33$**
2	**$K_2 = 0.67$**	**$П_2 = 0.28$**	**$M_2 = 0.67$**
3	**$K_3 = 1$**	**$П_3 = 0.43$**	**$M_3 = 1$**
4	**----------**	**$П_4 = 0.57$**	**----------**
5	**----------**	**$П_5 = 0.71$**	**----------**
6	**----------**	**$П_6 = 0.86$**	**----------**
7	**----------**	**$П_7 = 1$**	**----------**

Πίνακας 1: *Πίνακας Αποχρώσεων*

2.5 Κωδικοποίηση χαρακτήρα σε Matrix Barcode στο Unity

Εφόσον έχουν οριστεί πλέον οι τιμές που μπορεί να πάρει κάθε χρώμα, μένει να γίνει η μετατροπή ενός τυχαίου ASCII 7-bit χαρακτήρα στο αντίστοιχό του χρώμα.

a. Αρχικά μετατρέπουμε τον ASCII χαρακτήρα στη μοναδική φυσική τιμή του (στο δυαδικό ανάπτυγμα). Δηλαδή

$$Ψηφίο\ ASCII \leftrightarrow φυσικό\ b \in [0,127] \leftrightarrow b = \sum_{i=0}^{6} b_i 2^i = (b_6, b_5, b_4, b_3, b_2, b_1, b_0)$$

b. Στη συνέχεια, γίνεται ο υπολογισμός των τιμών των bit που αφορούν το κάθε χρώμα. Για την "εξαγωγή" των συνεχόμενων μπλοκ των bits που μας ενδιαφέρουν χρησιμοποιούμε την ακέραια διαίρεση του byte με το 2 και τις δυνάμεις του.

Υπενθυμίζουμε εδώ ότι με τον υπολογισμό του υπολοίπου της ακέραιας διαίρεσης του byte με το 2^b, γίνεται απόρριψη όλων των bit μετά τη θέση b, δηλαδή των bit στις θέσεις b+1, b+2 κλπ, ενώ με την ακέραια διαίρεση του byte με το 2^b, γίνεται απόρριψη των πριν τη θέση b, δηλαδή των bit στις θέσεις b-1, b-2, b-3, κλπ.

Από την παράγραφο 2.3 και την παραπάνω ισοδυναμία στο (a) της σελ. 18, θυμόμαστε ότι η δυάδα bits (b_1, b_0) αντιστοιχεί στο μπλε, η τριάδα bits (b_4, b_3, b_2) αντιστοιχεί στο πράσινο, ενώ η η δυάδα bits (b_6, b_5) αντιστοιχεί στο κόκκινο. Έτσι, χρησιμοποιώντας και τον πίνακα 1 στη σελίδα 18, μπορούμε να κωδικοποιήσουμε τους χαρακτήρες ASCII με μια τριάδα (R,G,B) που αντιστοιχεί σε μοναδική απόχρωση χρώματος.

Για παράδειγμα, έστω ότι θέλουμε να κωδικοποιήσουμε τον χαρακτήρα "z". Ο μοναδικός φυσικός αριθμός που αντιστοιχεί στο χαρακτήρα αυτό στην κωδικοποίηση ASCII 7-bit είναι ο 122 ή αλλιώς ο 1111010 σε δυαδική μορφή (βλέπε (a) παραπάνω). Υπολογίζοντας το υπόλοιπο της ακέραιας διαίρεσης με το 4, παίρνουμε την τιμή 2 ή δυαδικά (1,0) που είναι το μέρος του αριθμού που αντιστοιχεί στο μπλε. Αντίστοιχα υπολογίζονται και τα υπόλοιπα χρώματα.

Αν λοιπόν συμβολίσουμε με *b* το φυσικό αριθμό που αντιστοιχεί σε χαρακτήρα ASCII, τότε έχουμε

$$ΜΠΛΕ : \left(b \bmod \left(2^2\right)\right) \operatorname{div} \left(2^0\right) \leftrightarrow M_j, \text{για κάποιο στοιχείο της τρίτης στήλης του πίνακα } 1$$

$$ΠΡΑΣΙΝΟ : \left(b \bmod \left(2^5\right)\right) \operatorname{div} \left(2^2\right) \leftrightarrow Π_i, \text{για κάποιο στοιχείο της δεύτερης στήλης του πίνακα } 1$$

$$ΚΟΚΚΙΝΟ : \left(b \bmod \left(2^7\right)\right) \operatorname{div} \left(2^5\right) \leftrightarrow K_i, \text{για κάποιο στοιχείο της πρώτης στήλης του πίνακα } 1$$

Έτσι στο παραπάνω παράδειγμα έχουμε:

```
1 1 1 1 0 1 0 = 122

Bit        7    6    5    4    3    2    1    0
Data       -    1    1    1    1    0    1    0
(R = red, G = green, B = blue)

Blue → 122 mod 2^2 div 2^0 = 122 mod 4 = 2 → blue color = 0.6667

Green → 122 mod 2^5 div 2^2 = 26 div 4 = 6 → green color = 0.857

Red → 122 mod 2^7 div 2^5 = 122 div 32 = 3 → red color = 1
```

2.6 Εγκυρότητα μηνύματος

Έχοντας θέσει τους βασικούς κανόνες της κωδικοποίησης που περιγράφουν τις αντιστοιχίες κάθε ASCII 7-bit χαρακτήρα σε μοναδικά χρώματα, πρώτα κάνουμε έλεγχο του μηνύματος προς κωδικοποίηση ως προς την εγκυρότητά του. Η εγκυρότητα του μηνύματος, στην προκειμένη περίπτωση, εξαρτάται από δύο παράγοντες: το μέγεθός του και τους χαρακτήρες που περιέχει. Για τον παραπάνω έλεγχο χρησιμοποιούνται δύο boolean (δυαδικός τύπος δεδομένων) μεταβλητές οι οποίες περιγράφουν τις καταστάσεις αυτές παίρνοντας τις τιμές true/false (αληθές/ψευδές).

Η εγκυρότητα του μεγέθους του μηνύματος εξαρτάται από δύο συνθήκες:

1. Το μήνυμα δεν πρέπει να είναι κενό (null).
2. Το μήνυμα δεν πρέπει να είναι μεγαλύτερο από τους 899 χαρακτήρες.

Όσον αφορά στο (1) το μήνυμα δεν πρέπει να είναι κενό καθώς δεν υφίσταται το ίδιο το μήνυμα. Σχετικά με το (2), το μήνυμα δεν πρέπει να ξεπερνά το όριο των 899 χαρακτήρων. Το όριο αυτό εισήχθη για τον περιορισμό του χρόνου κωδικοποίησης και την υποβοήθηση των πειραματισμών. Για τον επιτυχή έλεγχο λοιπόν του μεγέθους πρέπει να είναι αληθής η παρακάτω λογική πράξη:

valid_size = not_null && (size_of_message < 899)

όπου valid_size είναι η boolean μεταβλητή που περιγράφει το αποδεκτό μέγεθος.

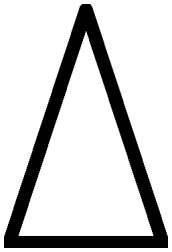
Ένας δεύτερος έλεγχος αφορά στο περιεχόμενο του μηνύματος, στο σκέλος του αν αυτό περιέχει μόνον ASCII 7-bit χαρακτήρες. Το αποτέλεσμα αυτού του ελέγχου αποθηκεύεται σε μια boolean μεταβλητή, συνολικά is_ASCII.

Τελικά το μήνυμα περνάει επιτυχώς τον έλεγχο εγκυρότητας αν η παρακάτω λογική πράξη είναι αληθής:

string_is_valid = is_ASCII && valid_size

2.7 Δημιουργία και τοποθέτηση του στοιχείου του Matrix Barcode στο Unity

Έχοντας ελέγξει το μήνυμα επιτυχώς και έχοντας κωδικοποιήσει κάθε χαρακτήρα, μένει μόνο η δημιουργία του βασικού στοιχείου του Matrix Barcode και η σωστή τοποθέτησή του στο χώρο. Ως στοιχείο του Matrix Barcode που κατασκευάσαμε, επιλέχθηκε το παρακάτω ισοσκελές τρίγωνο. Το συγκεκριμένο τρίγωνο δημιουργήθηκε στο Microsoft Paint των Windows. Θεωρητικά, η εφαρμογή λειτουργεί με οποιοδήποτε είδος ισοσκελούς τριγώνου.

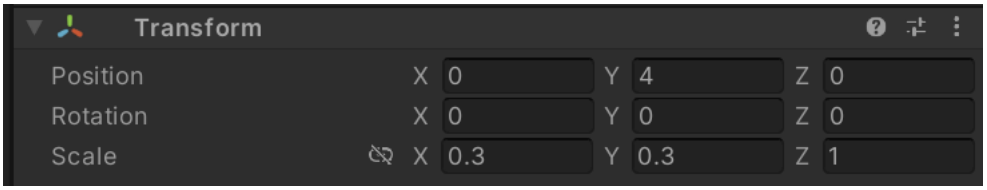


Σχήμα 14: *Το στοιχείο του Matrix Barcode*

Για κάθε χαρακτήρα που υπάρχει στο μήνυμα, παράγουμε ένα τρίγωνο το οποίο και χρωματίζουμε με βάση την κωδικοποίηση της παραγράφου 2.5 και το τοποθετούμε στην κατάλληλη θέση έτσι ώστε να σχηματιστεί ένα μεγαλύτερο όμοιο ισοσκελές τρίγωνο. Η τοποθέτηση του τριγώνου στην κατάλληλη θέση προϋποθέτει τη δημιουργία ενός προκατασκευασμένου αντικειμένου στο Unity (prefab GameObject). Το πλεονέκτημα τέτοιων αντικειμένων είναι ότι μπορούμε μέσω μιας εντολής να τα αντιγράψουμε και να τα εμφανίσουμε στον χρήστη, δηλαδή στο περιβάλλον της εφαρμογής. Ένα τέτοιου τύπου αντικείμενο μπορεί να έχει πολλές διαφορετικές ιδιότητες. Οι ιδιότητες στις οποίες αναφερόμαστε είναι τα “component” του Unity. Εμείς χρησιμοποιούμε κυρίως:

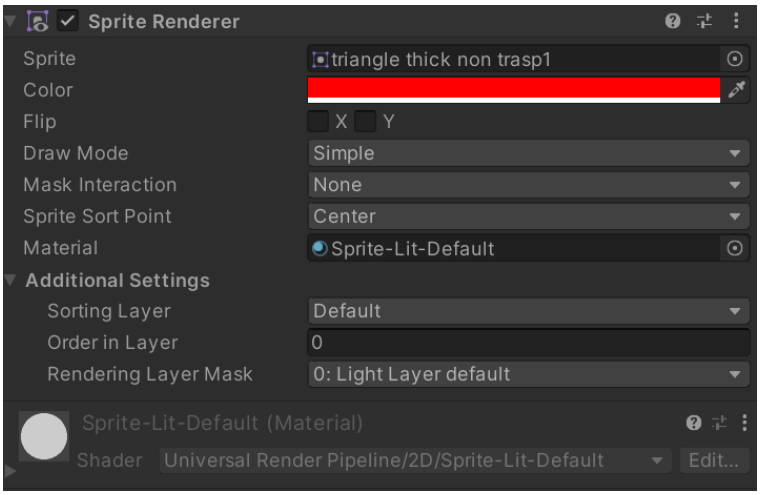
a) Την ιδιότητα “μετασχηματισμού” του Unity ή αλλιώς Transform. Το Transform αποτελεί αναπόσπαστο στοιχείο ενός Game Object και περιέχει τις καρτεσιανές συντεταγμένες του αντικειμένου στο χώρο, την περιστροφή του γύρω από τους άξονες στον εικονικό

τρισδιάστατο χώρο του προγράμματος καθώς και το σχετικό του μέγεθος. Εφόσον το Matrix Barcode είναι μια δισδιάστατη οντότητα, χρησιμοποιούμε μόνο το επίπεδο των αξόνων (x, y) και ως παρατηρητές βρισκόμαστε στον θετικό ημιάξονα του z.



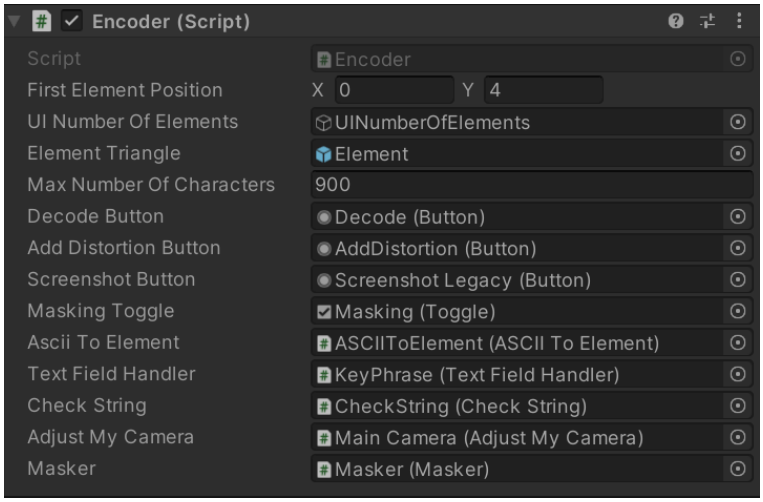
Σχήμα 15: *Το transform component του πρώτου στοιχείου του Unmasked Matrix Barcode. Τα πεδία Position, Rotation και Scale αφορούν στη θέση, στην περιστροφή και στην αναλογία του μεγέθους του αντικειμένου σε σχέση με το αρχικό για κάθε άξονα.*

b) Την ιδιότητα απεικόνισης δισδιάστατων γραφικών ή αλλιώς Sprite Renderer. Το Sprite Renderer δίνει τη δυνατότητα στο χρήστη να φορτώσει ένα γραφικό πάνω στο αντικείμενο του Unity και στη συνέχεια να του δώσει την απόχρωση που θέλει ο χρήστης. Το Sprite Renderer παρέχει κι άλλες δυνατότητες οι οποίες όμως δε θα μας απασχολήσουν.



Σχήμα 16: *Το Sprite Renderer Component ενός στοιχείου του Matrix Barcode. Το πεδίο Sprite περιέχει το αρχείο εικόνας που χρησιμοποιείται και το πεδίο Color το χρώμα που θα αποδοθεί στην εικόνα.*

c) Την ιδιότητα προσθήκης Script που επιτρέπει την προσθήκη και συγγραφή κώδικα που αφορά στο αντικείμενο αυτό.



Σχήμα 17: *Το Script Component του αντικειμένου της κωδικοποίησης. Τα πεδία του έχουν δημιουργηθεί μέσω κώδικα και περιέχουν μεταβλητές ή αναφορές σε άλλα χρήσιμα αντικείμενα.*

Η γεωμετρική μας κατασκευή διέπεται από τους εξής κανόνες:

- Ως πρώτο στοιχείο, θεωρούμε το στοιχείο του Matrix Barcode που βρίσκεται στην κορυφή (απέναντι από τη βάση) του τελικού τριγώνου που θα σχηματιστεί. Οι συντεταγμένες αυτού του στοιχείου στο περιβάλλον, ορίζονται από τον κώδικα του κωδικοποιητή και είναι (0,4) όπως φαίνεται και στο Σχήμα 17 (πεδίο First Element Position). Εφόσον κατασκευάζουμε το τρίγωνο από πάνω προς τα κάτω, ανεβάζοντας την αρχική θέση του πρώτου τριγώνου κατά 4 μονάδες στον άξονα y, γίνεται καλύτερη τοποθέτηση του κώδικα στο κέντρο της οθόνης.
- Κάθε γραμμή στοιχείων περιέχει 2(N-1)+1 στοιχεία όπου N ο αριθμός της γραμμής. Σε κάθε γραμμή περιστρέφουμε γύρω από τον άξονα z κατά 180° κάθε στοιχείο που ικανοποιεί τη συνθήκη

  $$v \bmod 2 = 0$$

  όπου v η σειρά εμφάνισης του στοιχείου στη γραμμή.
- Γνωρίζοντας το πλήθος των στοιχείων M που θα κωδικοποιηθούν και βάσει του γεγονότος ότι σε κάθε γραμμή N έχουμε 2N-1 στοιχεία μπορούμε να βρούμε πόσες γραμμές χρειαζόμαστε, λύνοντας την ανίσωση για το μικρότερο δυνατό N:

  $$1+3+5+\ldots+2N-1=N^2\geq M$$

- Χρησιμοποιώντας μια εντολή στο Unity, βρίσκουμε τις διαστάσεις του στοιχείου του Matrix Barcode (μήκος βάσης και ύψος του τριγώνου) και συνθέτουμε το Τριγωνικό Matrix Barcode με αναφορά στις συντεταγμένες του πρώτου στοιχείου. Έτσι, αν θεωρήσουμε (x_0, y_0) τις συντεταγμένες του πρώτου στοιχείου, ($x_κ$, $y_κ$) τις συντεταγμένες του στοιχείου κ για

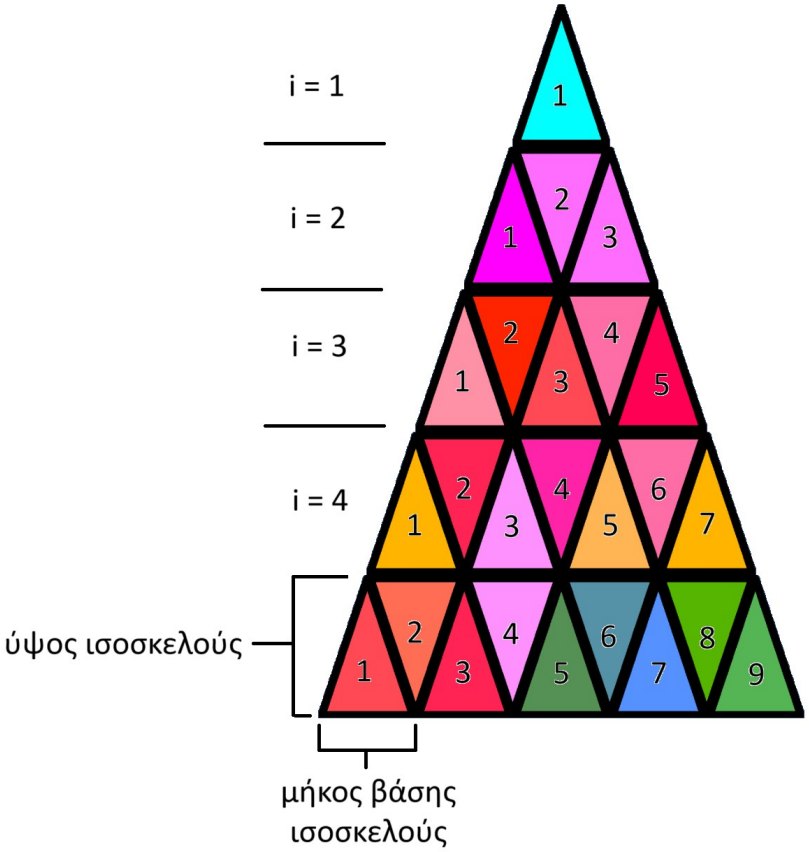
κ = 1, 2, 3…, i = 1, 2, 3... τον αριθμό γραμμής και j = 1, 2, 3... τη σειρά εμφάνισης του στοιχείου στη γραμμή, προκύπτει:

$$x_{\kappa} = x_0 + (j - i)\frac{\mu\acute{\eta}\kappa o\varsigma\ \beta\acute{\alpha}\sigma\eta\varsigma\ \iota\sigma o\sigma\kappa\varepsilon\lambda o\upsilon\varsigma}{2}$$

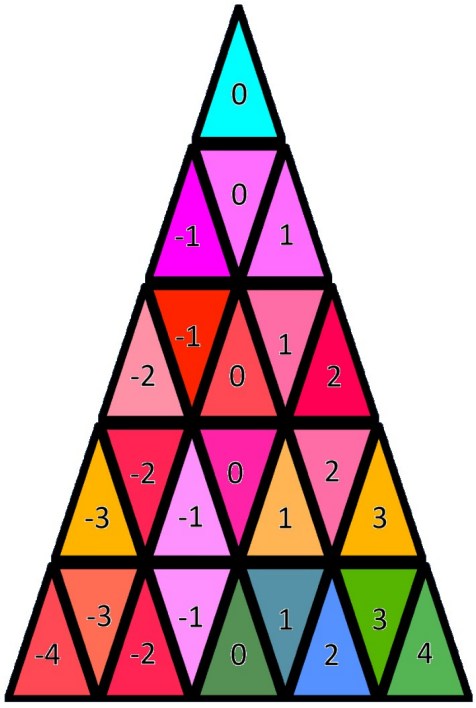
και

$$y_{\kappa} = y_0 - (i-1) * \acute{\upsilon}\psi o\varsigma\ \iota\sigma o\sigma\kappa\varepsilon\lambda o\upsilon\varsigma$$

Οι συγκεκριμένες σχέσεις γίνονται προφανείς από τις παρακάτω εικόνες όπου στο Σχήμα 18 κάθε στοιχείο περιέχει τον αριθμό j και στο Σχήμα 19 κάθε στοιχείο περιέχει τον όρο (j – 1)



Σχήμα 18: *Απεικόνιση των μεταβλητών που χρησιμοποιούνται στις παραπάνω σχέσεις. Μέσα σε κάθε τρίγωνο υπάρχει η τιμή του j.*



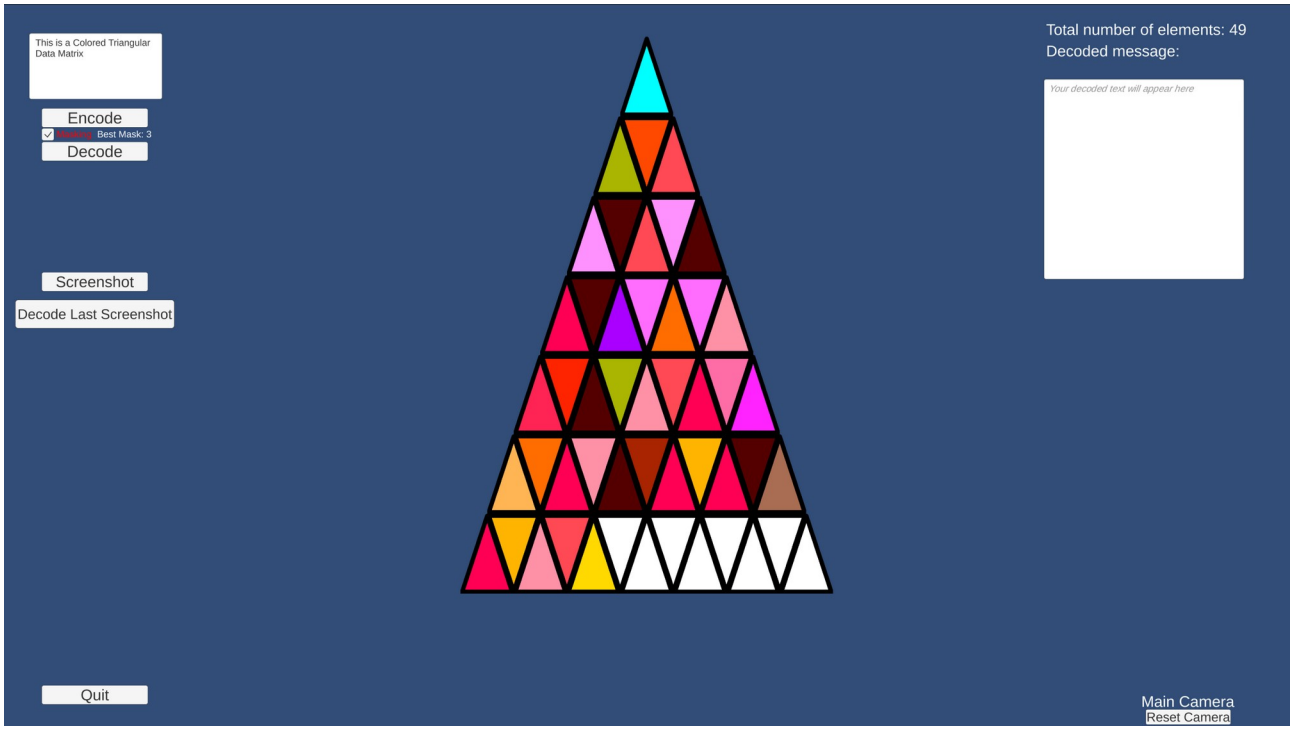
Σχήμα 19: *Κάθε τρίγωνο περιέχει την τιμή του όρου (j - i) όπου j η σειρά εμφάνισης του στοιχείου στη γραμμή και i ο αριθμός της γραμμής.*

Οι συντεταγμένες του κάθε στοιχείου κ αποθηκεύονται σε έναν πίνακα συντεταγμένων με το αντίστοιχο δείκτη τους για την αποκωδικοποίηση. Για παράδειγμα, το πρώτο στοιχείο με δείκτη 0 έχει συντεταγμένες (0, 4).

- Εφόσον έχουμε υπολογίσει τις ακριβείς θέσεις κάθε τριγώνου, παράγουμε τρίγωνα για κάθε χαρακτήρα του μηνύματος.
- Το πρώτο τρίγωνο χρησιμοποιείται για masking. Αναθέτουμε (αυθαίρετα) το κυανό χρώμα και συνεχίζουμε στο δεύτερο τρίγωνο με την κωδικοποίηση του πρώτου χαρακτήρα του μηνύματος και ούτω καθεξής.
- Αν το πλήθος των στοιχείων δεν είναι αρκετό για να καλυφθεί πλήρως η τελευταία γραμμή του τελικού τριγώνου, παράγουμε κενά στοιχεία μέχρι να συμπληρωθεί πλήρως η γραμμή.
- Στο τέλος αυτής της διαδικασίας αντιγράφουμε αυτό το αποτέλεσμα τέσσερις φορές σε ξεχωριστά σημεία στο χώρο για να εφαρμόσουμε τις μάσκες.

2.8 Επεξήγηση διεπαφής χρήστη και παράδειγμα στο Unity

Το λογισμικό, διαθέτει στο πάνω αριστερά μέρος της οθόνης ένα πεδίο κειμένου όπου ο χρήστης μπορεί να πληκτρολογήσει το μήνυμα προς αποκωδικοποίηση της επιλογής του. Ακριβώς από κάτω του υπάρχει το κουμπί του Encode το οποίο κωδικοποιεί το μήνυμα αυτό, εφόσον περάσει επιτυχώς τον έλεγχο εγκυρότητας. Με βάση αυτά που αναφέρθηκαν, το αποτέλεσμα της κωδικοποίησης του μηνύματος “This is a Colored Triangular Data Matrix” είναι το εξής:



Σχήμα 20: *Κωδικοποίηση μηνύματος στο Τριγωνικό Έγχρωμο Matrix Barcode*

Παρατηρούμε ότι το πρώτο τρίγωνο έχει το αυθαίρετο χρώμα του κυανού που αναθέσαμε και τα λευκά τρίγωνα στο τέλος που σηματοδοτούν ότι το μήνυμα έχει ολοκληρωθεί. Επίσης, στο πάνω δεξιά μέρος της οθόνης εμφανίζεται ο συνολικός αριθμός των στοιχείων που περιέχει το συγκεκριμένο Matrix Barcode. Αυτός ο αριθμός υπολογίζεται από το τετράγωνο του πλήθους των γραμμών, τύπος που προκύπτει από το άθροισμα των πρώτων λ περιττών αριθμών:

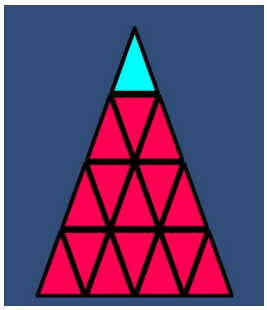
$$\sum_{\nu=0}^{\lambda}(2\nu+1)=(\lambda+1)^2$$

Κεφάλαιο 3. Masking (τεχνικές απόκρυψης)

Στο τρίτο κεφάλαιο περιγράφουμε στην πρώτη παράγραφο τους λόγους για τους οποίους χρησιμοποιείται το masking στο Matrix Barcode της παρούσης διπλωματικής. Στην παράγραφο 3.2 ορίζουμε τις μάσκες που χρησιμοποιούμε και περιγράφουμε τη λειτουργία τους. Ακολουθεί η παράγραφος 3.3 όπου εισάγουμε το σύστημα αξιολόγησης των μασκών και επιλέγουμε τη βέλτιστη με βάση την απόδοσή της. Η παράγραφος περιλαμβάνει και ένα παράδειγμα για την κατανόηση του συστήματος σε πρακτικό επίπεδο. Ο επίλογος του κεφαλαίου γίνεται στην παράγραφο 3.4 όπου παρέχουμε πληροφορίες για τον τρόπο χρήσης των μασκών στην εφαρμογή που κατασκευάστηκε καθώς και οδηγίες για τη χρήση της κάμερας της εφαρμογής.

3.1 Masking στο Matrix Barcode

Όπως αναφέρθηκε προηγουμένως, το masking αποτελεί μια τεχνική επικάλυψης και απόκρυψης των πληροφοριών. Η αναγκαιότητα του για την ενσωμάτωσή του σε ένα Matrix Barcode είναι παρόλα αυτά διαφορετική σε κάθε περίπτωση καθώς εξαρτάται από τη φύση και τη χρήση του Matrix Barcode. Στα ασπρόμαυρα Matrix Barcode για παράδειγμα είναι προφανές ότι, σε περίπτωση ύπαρξης συνεχόμενων bit ίδιων τιμών, θα παραχθεί ένας κώδικας ο οποίος είναι σε μεγάλα σημεία λευκός ή μαύρος. Αυτό δυσχεραίνει ιδιαίτερα την αποκωδικοποίηση από μια κάμερα σε πραγματικές συνθήκες επειδή η έλλειψη εναλλαγών μεταξύ των δύο χρωμάτων αποκρύπτει πληροφορίες όπως το μέγεθος του κάθε στοιχείου και η περιστροφή του. Για παράδειγμα, ένα QR Code στο οποίο υπάρχει μια περιοχή 8x8 όπου όλα τα στοιχεία του είναι μαύρα, κάνει σχεδόν αδύνατη την αξιολόγηση του μεγέθους του στοιχείου καθώς δεν υπάρχει κάποιο είδος περιγράμματος σε αυτά που σηματοδοτεί την “έναρξη” και τη “λήξη” ενός στοιχείου. Η εισαγωγή ενός πλήθους μασκών και η σύγκριση μεταξύ τους μέσω ενός συστήματος αξιολόγησης βοηθάει στην αποφυγή τέτοιου είδους καταστάσεων.



Σχήμα 21: *Η κωδικοποίηση του μηνύματος "aaaaaaaaaaaaaaaa" χωρίς μάσκα και το πρόβλημα που εμφανίζεται*

Επιστρέφοντας στο Matrix Barcode της παρούσης διπλωματικής, το masking κρίθηκε απαραίτητο για την αποφυγή παρόμοιων χρωμάτων όταν επαναλαμβάνονται χαρακτήρες, για την απόκρυψη πληροφοριών όταν αυτό είναι επιθυμητό καθώς και για την αισθητική αναβάθμιση του κώδικα.

Για τις ανάγκες του Matrix Barcode κατασκευάστηκαν τέσσερις διαφορετικές μάσκες που περιγράφονται παρακάτω.

3.2 Μάσκες του Matrix Barcode

Το βασικό χαρακτηριστικό κάθε μάσκας είναι το αναγνωριστικό της. Ως αναγνωριστικό της μάσκας έχουμε ορίσει ένα χρώμα το οποίο αποτυπώνεται στο πρώτο τρίγωνο του κώδικα και δεν επηρεάζεται από τους μετασχηματισμούς της ίδιας της μάσκας. Αυτό σημαίνει ότι η μάσκα δεν εφαρμόζεται σε αυτό το στοιχείο του κώδικα. Ο λόγος που το χρώμα αυτό πρέπει να παραμείνει αναλλοίωτο είναι για την επιτυχή αναγνώριση του είδους μάσκας κατά την αποκωδικοποίηση. Σε αντίθετη περίπτωση θα έπρεπε να αναγνωριστεί η ταυτότητα της μάσκας αφού πρώτα δοκιμαστεί ο άγνωστος και αντίστροφος μετασχηματισμός που ορίζει. Σημειώνουμε επίσης ότι τα λευκά στοιχεία που δηλώνουνε το τέλος του μηνύματος δεν επηρεάζονται από τις μάσκες. Έτσι κατασκευάστηκαν οι παρακάτω τέσσερις μάσκες.

Μάσκα 1: Η πρώτη μάσκα έχει ως αναγνωριστικό χρώμα το κόκκινο. Ο κανόνας της συγκεκριμένης μάσκας σε ψευδοκώδικα είναι ο εξής:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε BGR
}
```

Αυτό σημαίνει ότι για κάθε στοιχείο του Matrix Barcode, πέρα από το πρώτο (με δείκτη 0), η μάσκα θα μεταφέρει την τιμή του κόκκινου χρώματος του στοιχείου στο μπλε και το αντίστροφο.

Μάσκα 2: Η δεύτερη μάσκα έχει ως αναγνωριστικό χρώμα το πράσινο. Ο κανόνας της συγκεκριμένης μάσκας σε ψευδοκώδικα είναι ο εξής:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε RBG
}
```

Αυτό σημαίνει ότι για κάθε στοιχείο του Matrix Barcode που εφαρμόζεται η μάσκα, μεταφέρουμε την τιμή του πράσινου χρώματος του στοιχείου στο μπλε και το αντίστροφο.

Μάσκα 3: Η τρίτη μάσκα έχει ως αναγνωριστικό χρώμα το μπλε. Ο κανόνας της μάσκας 3 σε ψευδοκώδικα είναι ο εξής:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε GBR
}
```

Η μάσκα αυτή μετατοπίζει κυκλικά κατά αριστερά όλα τα χρώματα στο διάνυσμα χρώματος.

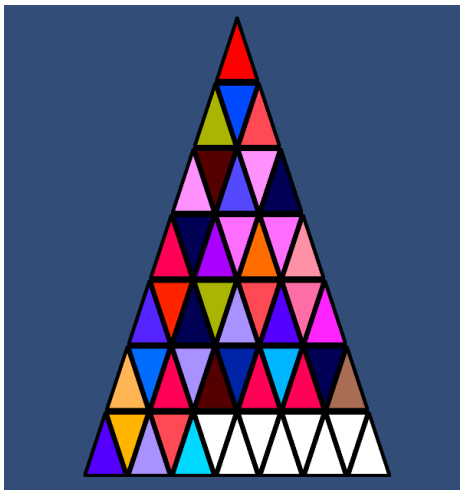
Μάσκα 4: Η τέταρτη μάσκα που κατασκευάστηκε έχει ως αναγνωριστικό χρώμα το λευκό. Ο κανόνας της μάσκας 4 σε ψευδοκώδικα είναι ο εξής:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε BRG
}
```

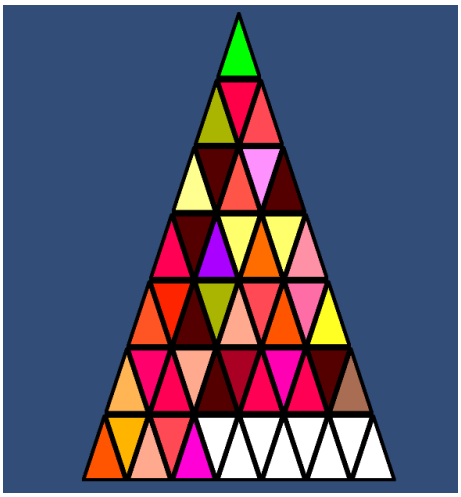
Η μάσκα αυτή μετατοπίζει κυκλικά κατά δεξιά όλα τα χρώματα στο διάνυσμα χρώματος.

Είναι προφανές ότι ως μάσκα μπορεί να χρησιμοποιηθεί οποιαδήποτε μετατροπή η οποία είναι αναστρέψιμη. Για παράδειγμα, θα μπορούσαν να χρησιμοποιηθούν κι άλλοι συνδυασμοί των RGB ή συγκεκριμένες προσθέσεις και αφαιρέσεις στα χρώματα, αρκεί να μην αλλοιώνεται και χάνεται η πληροφορία. Ωστόσο, η επιλογή της συνθήκης εφαρμογής της μάσκας, η οποία είναι κοινή σε όλες, έγινε σκόπιμα και έχει ως σκοπό την αποφυγή ίδιων χρωμάτων σε συνεχόμενα στοιχεία όταν ο χρήστης πληκτρολογήσει σε σειρά τουλάχιστον δύο ή περισσότερους ίδιους χαρακτήρες.

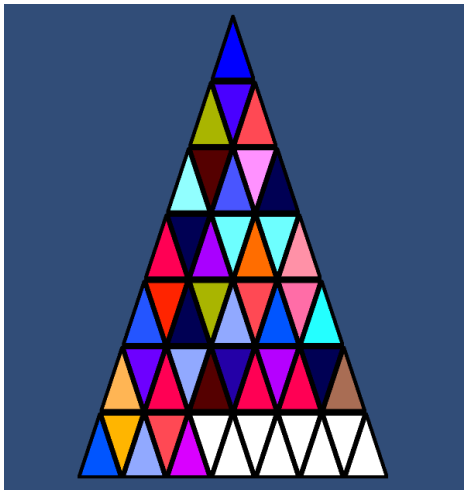
Έτσι το παράδειγμα του Σχήματος 20 του κεφαλαίου 2 παίρνει την εξής μορφή χρησιμοποιώντας τις τέσσερις διαφορετικές μάσκες για το μήνυμα “This is a Colored Triangular Data Matrix”:



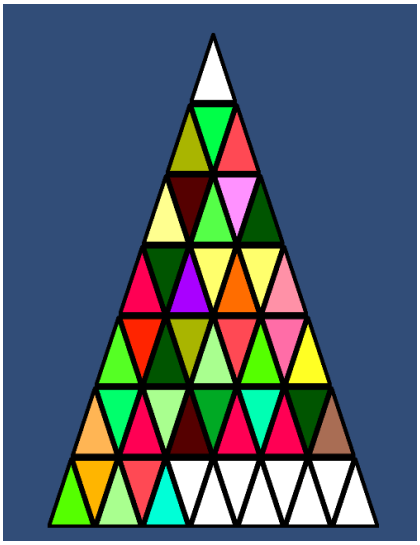
Σχήμα 22: *Αποτέλεσμα Μάσκας 1*



Σχήμα 23: *Αποτέλεσμα Μάσκας 2*



Σχήμα 24: *Αποτέλεσμα Μάσκας 3*



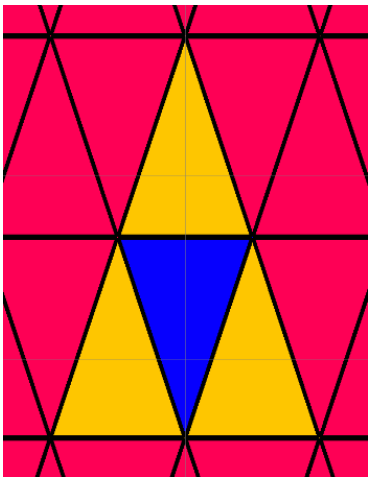
Σχήμα 25: *Αποτέλεσμα Μάσκας 4*

3.3 Σύστημα Αξιολόγησης (Scoring System)

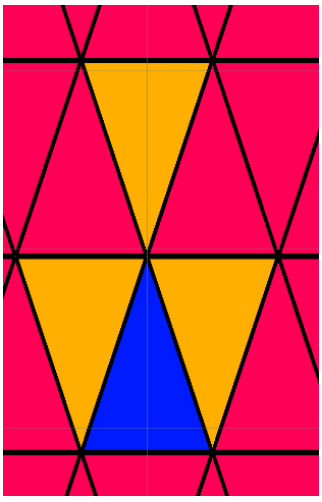
Εφόσον έχει κατασκευαστεί κάθε μάσκα και πλέον έχουμε μια οπτική εποπτεία του αποτελέσματος, η εφαρμογή πρέπει να αποφασίσει ποια μάσκα είναι καλύτερη, χρησιμοποιώντας ένα αυτοματοποιημένο σύστημα. Για τη λήψη αυτής της απόφασης κατασκευάσαμε ένα σύστημα αξιολόγησης κάθε μάσκας. Το σύστημα λειτουργεί ως εξής:

Κατ' αρχάς, θέτουμε την τιμή της βαθμολογίας κάθε μάσκας ίση με 0. Για κάθε στοιχείο $\alpha_{i,j}$ του Matrix Barcode εξετάζεται η ύπαρξη του αριστερού στοιχείου $\alpha_{i-1,\ j}$, του δεξιά στοιχείου $a_{i+1,\ j}$ και του από πάνω γειτονικού στοιχείου $\alpha_{i-1,\ j-1}$. Στη συνέχεια υπολογίζεται η ευκλείδια απόσταση του διανύσματος των χρωμάτων του, δηλαδή – αν q, v στοιχεία, τότε η χρωματική απόκλιση ισούται με:

$$Απόκλιση\ χρωμάτων = \sqrt{(R_q - R_v)^2 + (G_q - G_v)^2 + (B_q - B_v)^2}$$



Σχήμα 26: *Πρώτη περίπτωση γειτνίασης. Το μπλε στοιχείο συνορεύει με τα πορτοκαλί του σχήματος.*



Σχήμα 27: *Δεύτερη περίπτωση γειτνίασης. Το μπλε στοιχείο συνορεύει με τα πορτοκαλί του σχήματος.*

Κατά αυτόν τον τρόπο, η απόκλιση χρωμάτων μπορεί να πάρει τιμές στο κλειστό διάστημα $[0, \sqrt{3}]$. Στη συνέχεια:

- Αν η τιμή είναι μικρότερη του 0.5, η βαθμολογία μειώνεται κατά 5 καθώς τα χρώματα μοιάζουν.
- Αν η τιμή είναι μεγαλύτερη του 1, η βαθμολογία αυξάνεται κατά 1, εφόσον τα χρώματα διαφέρουν μεταξύ τους.
- Αν η τιμή είναι μεταξύ του 0.5 και του 1, δε συμβαίνει καμία αλλαγή στη βαθμολογία.

Στο τέλος της διαδικασίας συγκρίνονται οι βαθμολογίες που συγκεντρώθηκαν και επιλέγεται από το σύστημα η μάσκα με την υψηλότερη βαθμολογία ως βέλτιστη για την κωδικοποίηση του συγκεκριμένου μηνύματος. Στο παραπάνω παράδειγμα λοιπόν των σχημάτων 21-25 οι βαθμολογίες των μασκών είναι οι εξής:

Μάσκα 1: -93

Μάσκα 2: -219

Μάσκα 3: -75

Μάσκα 4: -77

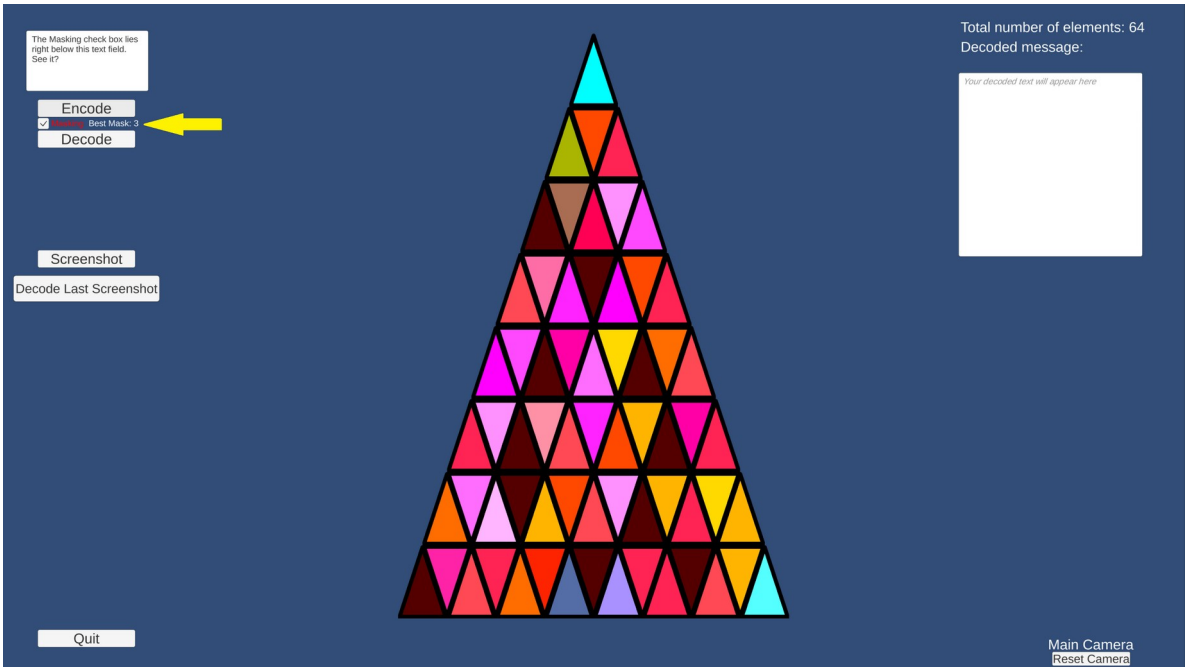
Κατά συνέπεια επιλέγεται η μάσκα 3 η οποία συγκέντρωσε τη μεγαλύτερη βαθμολογία.

3.4 Χρήση Masking στην εφαρμογή και κάμερα

Για να συμπεριληφθεί το Masking στη διαδικασία κωδικοποίησης, ο χρήστης χρειάζεται να έχει επιλεγμένο το “Masking” που βρίσκεται στο πάνω αριστερά μέρος της διεπαφής χρήστη. Η επιλογή αυτή είναι ενεργοποιημένη από προεπιλογή. Στη συνέχεια χρησιμοποιεί κανονικά την εφαρμογή για κωδικοποίηση όπως περιγράφηκε στο κεφάλαιο 2. Η μάσκα με την καλύτερη βαθμολογία θα εμφανιστεί στο πεδίο “Best Mask:”.

Με την ολοκλήρωση της κωδικοποίησης παράγονται πέντε συνολικά Matrix Barcode. Αυτό που φαίνεται πρώτο είναι το αποτέλεσμα της κωδικοποίησης χωρίς τη χρήση μάσκας. Αριστερά του βρίσκεται το αποτέλεσμα της πρώτης μάσκας, ακολουθεί το αποτέλεσμα της δεύτερης μάσκας και ούτω καθεξής. Για την πλοήγηση στο περιβάλλον χρησιμοποιείται ένα είδος “ελεύθερης κάμερας”. Η ελεύθερη κάμερα δίνει τη δυνατότητα στο χρήστη να μετακινηθεί στο χώρο για να εξετάσει τα αποτελέσματα της κωδικοποίησης χρησιμοποιώντας τις τέσσερις μάσκες που κατασκευάσαμε.

Η ελεύθερη κάμερα μπορεί να ελεγχθεί με τα βελάκια του πληκτρολογίου ή με τα πλήκτρα WASD για πάνω, αριστερά, κάτω και δεξιά αντίστοιχα. Πατώντας το πλήκτρο F9, κλειδώνουμε την κίνηση της κάμερας και την ξεκλειδώνουμε πατώντας το ξανά.



Σχήμα 28: *Στιγμιότυπο εικόνας από τη διεπαφή χρήστη. Το κίτρινο βέλος δείχνει στο σημείο του check box του masking.*

Κεφάλαιο 4. Αποκωδικοποίηση

Στο κεφάλαιο της αποκωδικοποίησης εξετάζουμε στην παράγραφο 4.1 τι σημαίνει αποκωδικοποίηση ενός Matrix Barcode και τι είδους κινδύνους ενέχει. Η παράγραφος 4.2 περιλαμβάνει τον πυρήνα της διαδικασίας ο οποίος αφορά στην αποκωδικοποίηση του Matrix Barcode σε ιδανικές συνθήκες χωρίς χρήση μάσκας και δημιουργείται η συνάρτηση αποκωδικοποίησης με μεταβλητές τα RGB χρώματα κάθε στοιχείου. Στη συνέχεια, στην παράγραφο 4.3 περιγράφουμε τα απαραίτητα βήματα για την αναγνώριση και αφαίρεση της μάσκας (unmasking) από ένα masked Matrix Barcode και την αποκωδικοποίησή του. Η παράγραφος 4.4 αποτελεί συνέχεια των παραδειγμάτων που παρουσιάστηκαν στις παραγράφους 2.8 και 3.4 και περιέχει χρήσιμες πληροφορίες για τη διεπαφή χρήστη της εφαρμογής.

4.1 Αποκωδικοποίηση Matrix Barcode

Η αποκωδικοποίηση ενός Matrix Barcode είναι η διαδικασία της μετατροπής των κωδικοποιημένων δεδομένων στην αρχική τους μορφή. Ουσιαστικά πρόκειται για την αντίστροφη διαδικασία της κωδικοποίησης. Ωστόσο, τα προβλήματα που παρουσιάζονται στην αποκωδικοποίηση διαφέρουν πολύ από αυτά της κωδικοποίησης. Ένα τεράστιο κομμάτι της αποκωδικοποίησης αποτελεί η επίλυση προβλημάτων που δημιουργούνται στη μεταφορά πληροφοριών μέσω ενός καναλιού με θόρυβο. Ο θόρυβος σε ένα κανάλι εισάγει παραμόρφωση στα δεδομένα που μεταφέρονται και δυσχεραίνει την επιτυχή αποκωδικοποίηση.

Στην περίπτωση της αποκωδικοποίησης ενός Matrix Barcode ειδικότερα, ο θόρυβος μπορεί να προκύψει από μια λανθασμένη ανάγνωση της τιμής ενός χρώματος (είτε από την κάμερα είτε από ανακρίβειες στις τιμές του RGB χρώματος), είτε από αλλοίωση πληροφορίας με οποιοδήποτε τρόπο (φθορά, κακή φωτεινότητα), είτε από μετασχηματισμό του Matrix Barcode όπως π.χ. είναι μια περιστροφή. Τα προβλήματα αυτά γίνονται ακόμα πιο έντονα όταν η διαδικασία της αποκωδικοποίησης γίνεται σε πραγματικές συνθήκες (όπως είναι το “σκανάρισμα” ενός QR Code αποτυπωμένου σε χαρτί από μια κάμερα κινητού) και όχι ψηφιακά (όπως είναι το “σκανάρισμα” του Cronto Visual Cryptogram που αναφέρθηκε στο κεφάλαιο 1).

Σχήμα 29: *Ένα παραμορφωμένο QR Code που είναι περιστραμμένο και κυρτό σε σημεία*

Τις περισσότερες φορές τα προβλήματα αυτά λύνονται χρησιμοποιώντας τεχνικές error correction κατά την κωδικοποίηση και αξιοποιώντας την πλεονάζουσα πληροφορία που υπάρχει κατά την αποκωδικοποίηση. Επίσης, πολύ βοηθητικά είναι και τα μοτίβα χρονισμού και ευθυγράμμισης που εξετάσαμε στο κεφάλαιο 1. Στην προσομοίωση του Matrix Barcode της παρούσας διπλωματικής, δε χρησιμοποιείται κάποια τεχνική error correction αλλά εξετάζεται αρχικά η λειτουργικότητα του Matrix Barcode που κατασκευάστηκε σε ιδανικές συνθήκες.

4.2 Αποκωδικοποίηση του Matrix Barcode χωρίς χρήση μάσκας

Κατά τη διάρκεια της κωδικοποίησης, δημιουργούνται συνολικά 5 αντικείμενα (Game Objects) στο Unity. Τα αντικείμενα αυτά είναι τα "γονικά αντικείμενα" (parent objects) της κάθε μάσκας και περιέχουν ως "παιδιά" τους (child objects) όλα τα στοιχεία του αντίστοιχου Matrix Barcode. Έτσι το πρώτο parent object έχει ως παιδιά του σειριακά ταξινομημένα όλα τα στοιχεία του unmasked Matrix Barcode, το δεύτερο parent object τα παιδιά της δεύτερης μάσκας κ.ο.κ.

Έχοντας αναφορά στο γονικό αντικείμενο που περιέχει ως παιδιά τα στοιχεία του unmasked Matrix Barcode, γίνεται εφικτή η πρόσβαση στο component Sprite Renderer (βλ. παράγραφο 2.7) του κάθε αντικειμένου – στοιχείου. Κατά συνέπεια, είναι δυνατή η σειριακή ανάγνωση των χρωμάτων του κάθε στοιχείου. Η διαδικασία ξεκινάει από το δεύτερο στοιχείο του Matrix Barcode καθώς το πρώτο περιλαμβάνει κανονικά την πληροφορία για την ταυτότητα της μάσκας.

Έστω, λοιπόν, ότι ένα τυχαίο στοιχείο του Matrix Barcode έχει χρώμα color = (r, g, b) και κάθε χρώμα έχει βάθος χρώματος r_d, g_d, και b_d αντίστοιχα. Τότε μπορούμε να υπολογίσουμε τον κωδικοποιημένο χαρακτήρα με βάση τους εξής τύπους:

$$R_{δεκαδικό} = r\left(2^{r_d} - 1\right) = 3r$$

$$G_{δεκαδικό} = g\left(2^{g_d} - 1\right) = 7g$$

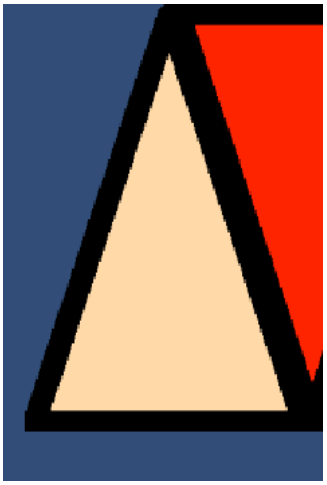
$$B_{δεκαδικό} = b * (2^{b_d} - 1) = 3b$$

$$Αριθμός\ χαρακτήρα = R_{δεκαδικό} 2^{g_d + b_d} + G_{δεκαδικό} 2^{b_d} + B_{δεκαδικό} 2^0 = 3r\,2^5 + 7g\,2^2 + 3b = 96r + 28g + 3b$$

$$Φυσικός\ αριθμός\ χαρακτήρα = Στρογγυλοποίηση(αριθμός\ χαρακτήρα)$$

Λόγω δεκαδικών ψηφίων και ατελών υπολογισμών ο φυσικός αριθμός χαρακτήρα στρογγυλοποιείται στον κοντινότερο φυσικό αριθμό. Αυτός ο αριθμός αντιστοιχεί σε έναν μοναδικό χαρακτήρα βάση του Σχήματος 10 της παραγράφου 2.1.

Για παράδειγμα, έστω ότι ένα στοιχείο του Matrix Barcode έχει την απόχρωση της παρακάτω εικόνας.



Σχήμα 30: *Απόχρωση παραδείγματος παραγράφου 4.1*

Αποκτώντας πρόσβαση στο Sprite Renderer του στοιχείου παίρνουμε τις τιμές χρωμάτων RGB (1, 0.857, 0.667). Χρησιμοποιώντας τους παραπάνω τύπους, έχουμε:

$$Αριθμός\ χαρακτήρα = 96r + 28g + 3b = 96 * 1 + 28 * 0.857 + 3 * 0.667 = 96 + 23.996 + 2.001 = 121.997$$

$$Φυσικός\ αριθμός\ χαρακτήρα = Στρογγυλοποίηση(αριθμός\ χαρακτήρα) = 122$$

Σύμφωνα με τον πίνακα ASCII, ο χαρακτήρας που αντιστοιχεί στο νούμερο 122 είναι ο χαρακτήρας “z”. Η αποκωδικοποίηση έγινε επιτυχώς.

4.3 Αποκωδικοποίηση του βέλτιστου masked Matrix Barcode

Όπως αναφέρθηκε στην παράγραφο 3.3, μετά την ολοκλήρωση της διαδικασίας απόκρυψης (masking) επιλέγεται από το σύστημα η βέλτιστη μάσκα σύμφωνα με τους κανόνες που ορίσαμε.

Η αποκωδικοποίηση του Matrix Barcode στο οποίο έχει χρησιμοποιηθεί masking άγνωστης ταυτότητας χωρίζεται σε τρία βήματα:

1. Εύρεση της ταυτότητας της μάσκας.
2. Εφαρμογή της αντίστροφης συνάρτησης μάσκας.
3. Αποκωδικοποίηση του unmasked Matrix Barcode ακολουθώντας τη διαδικασία που περιγράφηκε στην παράγραφο 4.2.

Όπως αναφέρθηκε στην παράγραφο 4.2, μπορούμε να αποκτήσουμε πρόσβαση στις αποχρώσεις των στοιχείων μέσω του component Sprite Renderer του κάθε αντικειμένου.

Αρχικά εξετάζουμε το χρώμα του πρώτου στοιχείου του Matrix Barcode καθώς αυτό έχει την πληροφορία της ταυτότητας της μάσκας. Κάνουμε αναζήτηση του χρώματος στον πίνακα που περιέχει τα 4 διαφορετικά χρώματα που έχουμε αποδώσει σε κάθε μάσκα μέχρι να βρούμε τη σωστή.

Γνωρίζοντας πλέον την ταυτότητα της μάσκας που χρησιμοποιήθηκε για την απόκρυψη των στοιχείων του Matrix Barcode, μπορούμε να εφαρμόσουμε την αντίστροφη συνάρτηση της μάσκας σε όλα τα στοιχεία του κώδικα (εκτός από το πρώτο), έτσι ώστε να καταλήξουμε στο unmasked Matrix Barcode. Οι αντίστροφες συναρτήσεις των μασκών για κάθε στοιχείο σε μορφή ψευδοκώδικα είναι οι εξής:

Μάσκα 1:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε BGR
}
```

Μάσκα 2:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε RBG
}
```

Μάσκα 3:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε BRG
}
```

Μάσκα 4:

```
ΑΝ (αριθμός_στοιχείου mod 2 == 0)
{
        ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ RGB σε GBR
}
```

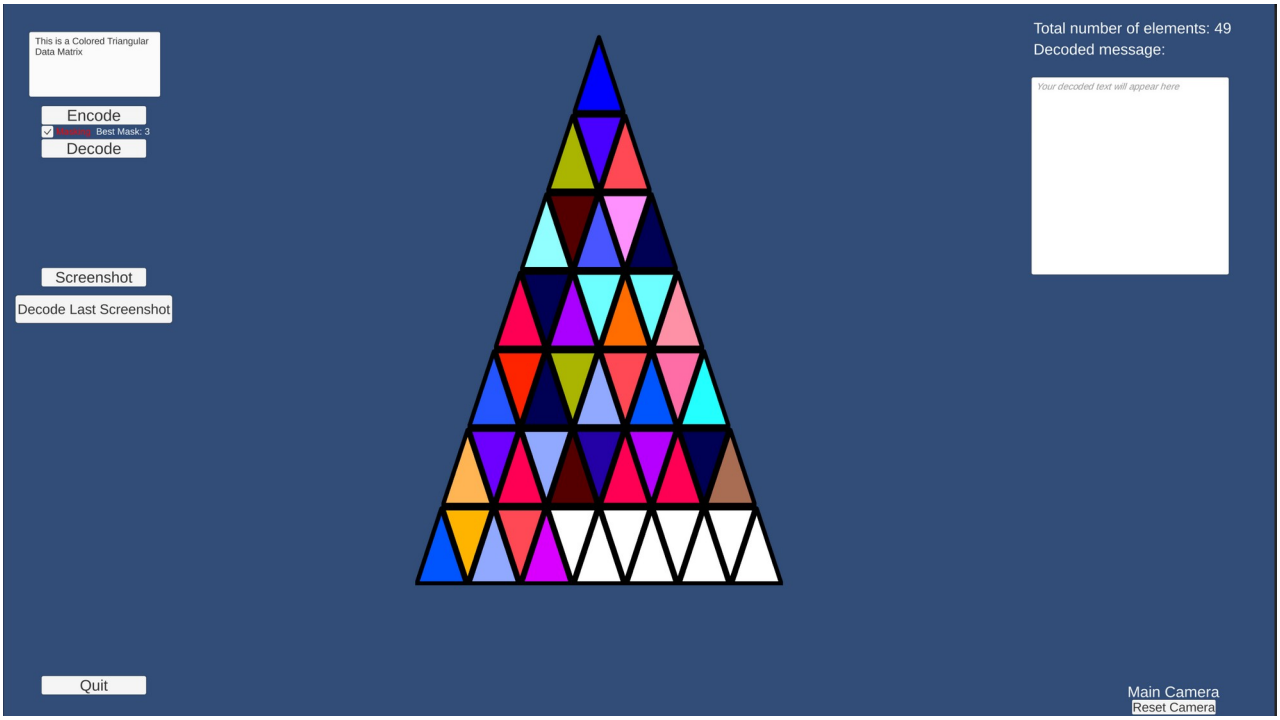
όπου ο αριθμός_στοιχείου παίρνει τιμές από 1 έως και το πλήθος των στοιχείων του Matrix Barcode.

Στο τέλος της διαδικασίας του unmasking, είναι αποθηκευμένο στο σύστημα το unmasked Matrix Barcode. Συνεπώς, ακολουθώντας τη διαδικασία που περιγράφηκε στην παράγραφο 4.2, ολοκληρώνουμε τη διαδικασία αποκωδικοποίησης του masked Matrix Barcode.

4.4 Παράδειγμα αποκωδικοποίησης και επεξήγηση διεπαφής στο Unity

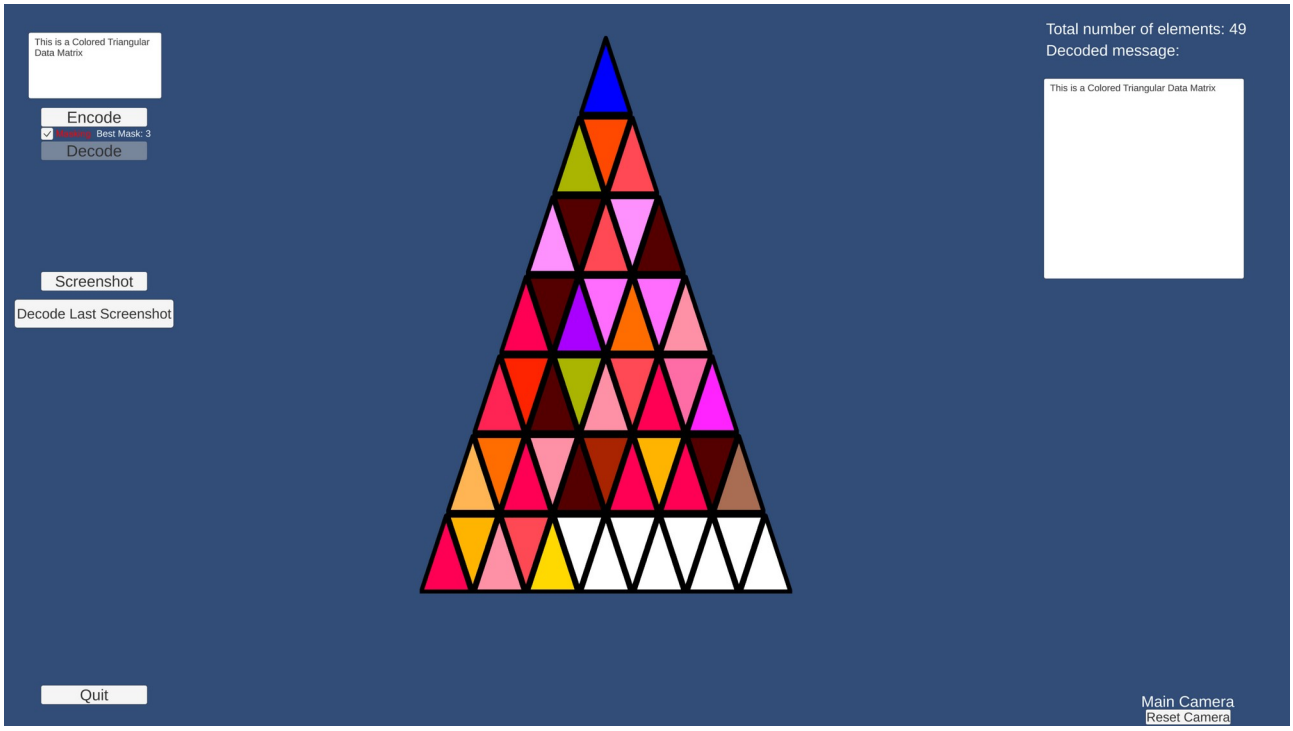
Συνεχίζοντας το παράδειγμα των παραγράφων 2.8 και 3.4, ο χρήστης έχει πληκτρολογήσει το κείμενο της επιλογής του και έχει κωδικοποιήσει το μήνυμα “This is a Colored Triangular Data Matrix”. Μετά το πάτημα του κουμπιού Encode (κωδικοποίηση), ενεργοποιείται το κουμπί Decode (αποκωδικοποίηση).

Όπως αναφέρθηκε στην παράγραφο 3.4, το σύστημα μας ενημερώνει (στο πεδίο “Best Mask: “) για τη βέλτιστη μάσκα σύμφωνα με το σύστημα αξιολόγησης, η οποία είναι η τρίτη. Χρησιμοποιώντας την ελεύθερη κάμερα, μετακινούμε τον παρατηρητή στο σωστό Matrix Barcode που είναι το τέταρτο κατά σειρά. Σε αυτό το σημείο, το στιγμιότυπο οθόνης που βλέπει ο χρήστης είναι το εξής:



Σχήμα 31: *Στιγμιότυπο οθόνης πριν την αποκωδικοποίηση. Ο χρήστης έχει μετακινήσει την κάμερα και ελέγχει το αποτέλεσμα του Matrix Barcode με χρήση της τρίτης μάσκας για το μήνυμα της επιλογής του. Το κουμπί Decode είναι ενεργοποιημένο στο πάνω αριστερά μέρος της διεπαφής χρήστη.*

Στη συνέχεια, πατώντας το κουμπί Decode, το Matrix Barcode της εικόνας γίνεται unmask, αποκωδικοποιείται και εμφανίζεται στο πεδίο κειμένου που βρίσκεται στο πάνω δεξιά μέρος της διεπαφής χρήστη. Το στιγμιότυπο οθόνης του χρήστη πλέον είναι το εξής:



Σχήμα 32: *Στιγμιότυπο οθόνης μετά την αποκωδικοποίηση. Οι αποχρώσεις του Matrix Barcode αντιστοιχούν πλέον στην εκδοχή του χωρίς μάσκα. Το πεδίο κειμένου στο πάνω δεξιά μέρος της οθόνης εμφανίζει το αποκωδικοποιημένο μήνυμα.*

Μετά το τέλος της επιτυχούς αποκωδικοποίησης, συμπεραίνουμε ότι το Matrix Barcode της παρούσης διπλωματικής είναι λειτουργικό σε ιδανικές συνθήκες όπου οι τιμές χρώματος κάθε στοιχείου είναι χωρίς καμία αμφιβολία πραγματικές και δεν αποκλίνουν (σημαντικά) από αυτές της κωδικοποίησης.

Κεφάλαιο 5. Αποκωδικοποίηση από Screenshot

Στο κεφάλαιο αυτό, και ειδικότερα στην παράγραφο 5.1, παρουσιάζουμε τις συνθήκες υπό τις οποίες θα γίνει η αποκωδικοποίηση του Matrix Barcode. Στη συνέχεια, στην παράγραφο 5.2, παρέχουμε πληροφορίες για τη λειτουργία “Screenshot” της εφαρμογής καθώς και για τις προδιαγραφές του στιγμιότυπου οθόνης που παράγεται. Στην παράγραφο 5.3 περιγράφουμε τη διαδικασία επεξεργασίας του στιγμιότυπου οθόνης που έχουμε έτσι ώστε να ανιχνεύσουμε τις ακμές του. Ακολουθεί η παράγραφος 5.4 όπου βρίσκουμε την ακριβή τοποθεσία του Matrix Barcode στο στιγμιότυπο οθόνης. Στην παράγραφο 5.5, η οποία αποτελεί και τον πυρήνα του συγκεκριμένου κεφαλαίου, αναλύουμε τη διαδικασία εύρεσης των συντεταγμένων του κάθε στοιχείου του Matrix Barcode. Στην παράγραφο 5.6 παρουσιάζουμε τη διαδικασία της αποκωδικοποίησης χρησιμοποιώντας όλα τα δεδομένα που έχουμε βρει και παρουσιάζουμε το τελικό αποτέλεσμα της αποκωδικοποίησης ενός Matrix Barcode από ένα στιγμιότυπο οθόνης.

5.1 Αποκωδικοποίηση Matrix Barcode χωρίς πρόσβαση στα δεδομένα της εφαρμογής

Έχοντας αποδείξει ότι το Matrix Barcode είναι λειτουργικό σε ιδανικές συνθήκες, επιχειρούμε να κάνουμε το επόμενο βήμα – αυτό της αποκωδικοποίησης από στιγμιότυπο οθόνης της εφαρμογής. Έχοντας μόνο ως δεδομένα ένα στιγμιότυπο οθόνης και απλά τους κανόνες που διέπουν την κωδικοποίηση (και άρα και την αποκωδικοποίηση), δημιουργούμε ένα πιο ρεαλιστικό περιβάλλον αποκωδικοποίησης καθώς η πληροφορία του Matrix Barcode μεταφέρεται κωδικοποιημένη σε μορφή εικόνας από τον πομπό στο δέκτη. Ο πομπός χρειάζεται να έχει μόνο τους κανόνες κωδικοποίησης του Matrix Barcode και ο δέκτης μόνο τους κανόνες αποκωδικοποίησης για την επιτυχή μετάδοση του μηνύματος.

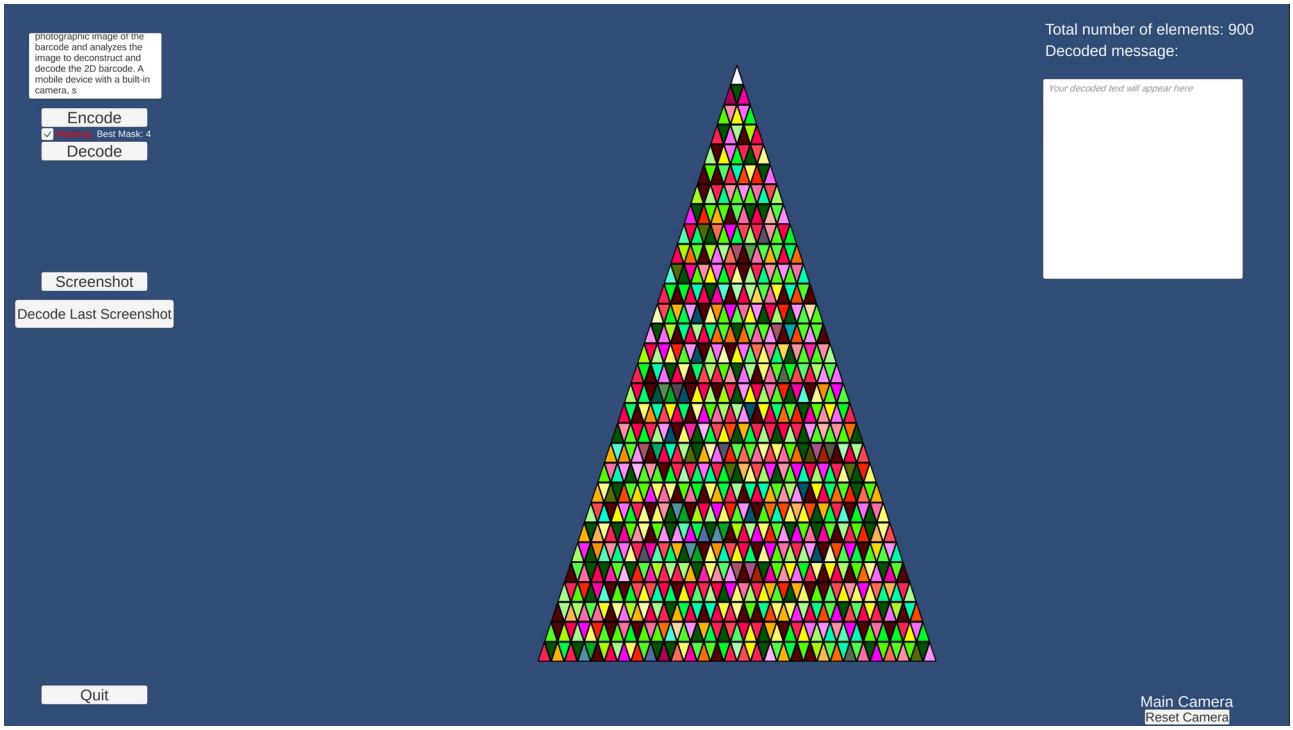
Στο προηγούμενο κεφάλαιο η ανάγνωση των τιμών των χρωμάτων γινόταν μέσα από την ίδια την εφαρμογή και άρα δεν υπήρχαν σοβαρές αποκλίσεις. Σε αυτή την περίπτωση επιχειρούμε να πάρουμε τις τιμές των χρωμάτων από ένα στιγμιότυπο οθόνης το οποίο είναι αποθηκευμένο σε μορφή αρχείου .png (Portable Network Graphics) στον υπολογιστή μας. Η εφαρμογή μας καλείται να βρει σε ουδέτερο φόντο τη θέση του Matrix Barcode, να δημιουργήσει τον πίνακα άγνωστου μεγέθους με τα στοιχεία αγνώστων διαστάσεων και στη συνέχεια να ολοκληρώσει την αποκωδικοποίηση με επιτυχία.

5.2 Εξαγωγή στιγμιότυπου οθόνης στο Unity

Το πρώτο βήμα που χρειάζεται να γίνει είναι να πραγματοποιηθεί η λήψη στιγμιότυπου εικόνας στην εφαρμογή ώστε να αποτυπωθεί το Matrix Barcode στο αρχείου .png που θέλουμε να επεξεργαστούμε.

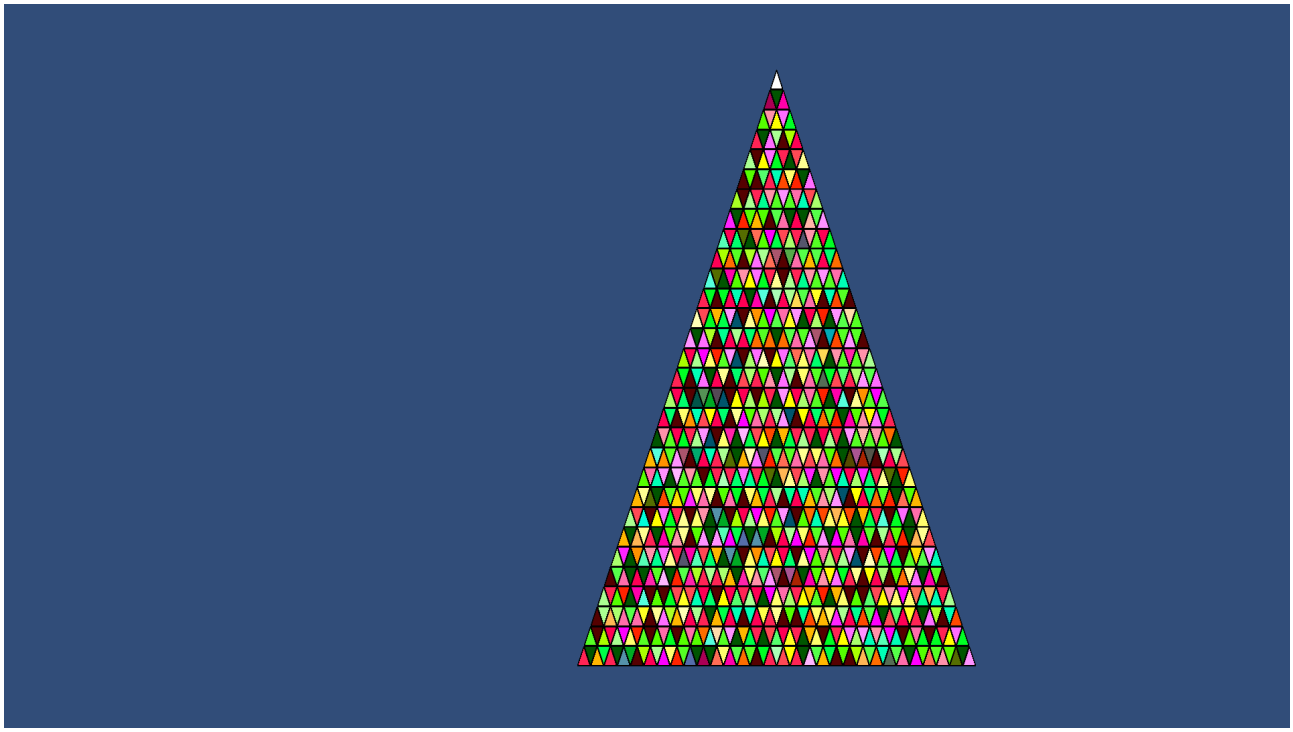
Προστέθηκε λοιπόν μια ακόμη κάμερα στην εφαρμογή με ανάλυση 1920x1080 η οποία χρησιμοποιείται για την εξαγωγή του screenshot. Η κάμερα αυτή δεν καταγράφει τη διεπαφή χρήστη και αποθηκεύει οτιδήποτε άλλο βλέπει σε μορφή “δισδιάστατης υφής”, δηλαδή σε μορφή της κλάσης Texture2D της Unity. Στη συνέχεια με τις κατάλληλες εντολές στο αντίστοιχο Script

Component (κώδικα), κωδικοποιούμε την υφή σε αρχείο .png και το αποθηκεύουμε στον υπολογιστή του χρήστη με όνομα αρχείου “SavedScreen.png”. Για να δημιουργήσουμε αυτό το screenshot, αφού κάνουμε την κωδικοποίηση και μετακινήσουμε την κάμερα στο επιθυμητό σημείο, πατάμε το κουμπί Screenshot στη διεπαφή χρήστη που βρίσκεται στο αριστερό μέρος της οθόνης. Στο παρακάτω παράδειγμα, έχουμε κωδικοποιήσει ένα μέρος κειμένου του λήμματος Barcode της ψηφιακής εγκυκλοπαίδειας Wikipedia.



Σχήμα 33: *Στιγμιότυπο οθόνης μαζί με τη διεπαφή χρήστη της τέταρτης μάσκας ενός κωδικοποιημένου μηνύματος.*

Στην εικόνα του Σχήματος 33, αν πατήσουμε το κουμπί Screenshot θα εξάγουμε το εξής στιγμιότυπο εικόνας διαστάσεων 1920 επί 1080 εικονοστοιχείων (pixel):



Σχήμα 34: *Στιγμιότυπο οθόνης διαστάσεων 1920x1080 του οποίου η λήψη έγινε χρησιμοποιώντας την εφαρμογή.*

Όπως φαίνεται στην εικόνα του Σχήματος 34, η κάμερα που χρησιμοποιήσαμε, δεν κατέγραψε τα κουμπιά και πεδία της διεπαφής χρήστη και ουσιαστικά έχει φωτογραφίσει το Matrix Barcode που κατασκευάσαμε σε ιδανικές συνθήκες σε μια τυχαία θέση στο επίπεδο x, y.

5.3 Διαδικασία επεξεργασίας του στιγμιότυπου οθόνης

Για την επιτυχή αποκωδικοποίηση του στιγμιότυπου οθόνης, είναι απαραίτητη πρώτα μια επεξεργασία του. Η επεξεργασία του αποσκοπεί στην εύρεση του κάθε στοιχείου - τριγώνου και κατ' επέκταση των συντεταγμένων όπου θα γίνει ανάγνωση της τιμής των χρωμάτων του. Όπως είναι φανερό, το παραπάνω αποτελεί κομμάτι της επιστήμης της Επεξεργασίας Εικόνας (Image Processing) και το πρώτο μας βήμα είναι να ανιχνεύσουμε τις ακμές στην εικόνα μας.

Συνήθως για την ανίχνευση ακμών χρησιμοποιούνται φίλτρα όπως το Λαπλασιανό φίλτρο ανίχνευσης ακμών. Στη συγκεκριμένη περίπτωση και για λόγους ταχύτητας, χρησιμοποιήθηκε αντί αυτού η παρακάτω λογική:

1. Γίνεται μετατροπή της εικόνας σε αποχρώσεις της κλίμακας του γκρι (grayscale). Για τη μετατροπή αυτή "σαρώνουμε" κάθε εικονοστοιχείο (pixel) της εικόνας και αποκτάμε πρόσβαση στο χρώμα του. Χρησιμοποιούμε την έτοιμη συνάρτηση του Unity για μετατροπή της απόχρωσης σε κλίμακα του γκρι η οποία για χρώματα τυχαίου εικονοστοιχείου $κ_{x,y}$ = (R, G, B) είναι η παρακάτω:

$$\text{Color}_{grayscale} = 0.299\,R + 0.587\,G + 0.114\,B$$

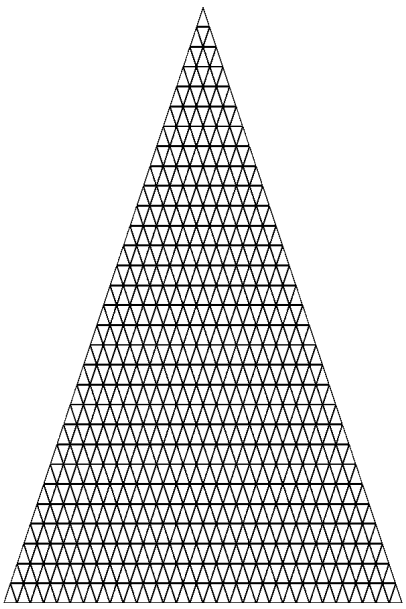
Όπως φαίνεται από την εξίσωση, το χρώμα κλίμακας του γκρι είναι ένας δεκαδικός αριθμός ο οποίος παίρνει τιμές από 0 για μαύρο έως και 1 για λευκό χρώμα.

2. Χρησιμοποιείται η παρακάτω συνάρτηση κατωφλίου σε μορφή ψευδοκώδικα:

```
        ΓΙΑ ΚΑΘΕ εικονοστοιχείο στην εικόνα
{
        ΑΝ (χρώμα_εικονοστοιχείου_σε_greyscale > 0.01 == 0)
        {
                Χρώμα_εικονοστοιχείου = λευκό
        }
        ΑΛΛΙΩΣ
        {
                Χρώμα_εικονοστοιχείου = μαύρο
        }
}
```

Σύμφωνα με τον ψευδοκώδικα αυτό, συγκρίνουμε το χρώμα σε κλίμακα γκρι του εικονοστοιχείου με την τιμή 0.01. Αν το χρώμα έχει τιμή μεγαλύτερη από 0.01 τότε αναθέτουμε το χρώμα λευκό σε αυτό. Σε περίπτωση που το χρώμα έχει τιμή μικρότερη ή ίση με 0.01 το χρωματίζουμε με μαύρο χρώμα. Στην ουσία η συνάρτηση βρίσκει τα πολύ σκούρα σημεία στην εικόνα και τα χρωματίζει με μαύρο χρώμα και χρωματίζει με λευκό χρώμα όλα τα υπόλοιπα.

Με το πέρας αυτής της διαδικασίας και για το παράδειγμα που εξετάζουμε στα Σχήματα 33 και 34, δημιουργούμε το παρακάτω αρχείο εικόνας το οποίο και αποθηκεύουμε στον υπολογιστή του χρήστη με όνομα “SavedScreenBlackAndWhite.png”.



Σχήμα 35: *Ασπρόμαυρη εικόνα μετά τη μετατροπή σε κλίμακα του γκρι και εφαρμογή συνάρτησης κατωφλίου*

5.4 Εύρεση χωρικών προδιαγραφών του Matrix Barcode σε στιγμιότυπο οθόνης

Έχοντας βρει τις ακμές του Matrix Barcode στην προηγούμενη παράγραφο, συνεχίζουμε την επεξεργασία της εικόνας αναζητώντας τις ακριβείς συντεταγμένες εμφάνισής του. Για την εύρεση του υψηλότερου σημείου του Matrix Barcode χρησιμοποιούμε την εξής λογική σε μορφή ψευδοκώδικα:

```
ΓΙΑ ΚΑΘΕ στήλη εικονοστοιχείου
        {
        ΓΙΑ ΚΑΘΕ γραμμή εικονοστοιχείου
                {
                        ΑΝ (χρώμα_εικονοστοιχείου == μαύρο)
                                {
                                        ΕΠΙΣΤΡΕΨΕ συντεταγμένες_εικονοστοιχείου
                                }
                }
        }
```

Εφόσον γνωρίζουμε ότι ο προσανατολισμός του είναι ορθός, καθώς και ότι το πρώτο μαύρο εικονοστοιχείο που θα βρούμε “σκανάροντας” την εικόνα από πάνω αριστερά προς τα δεξιά μπορούμε να βρούμε το υψηλότερο σημείο του Matrix Barcode.

Με γνωστές πλέον τις συντεταγμένες της κορυφής του τελικού τριγώνου και “σκανάροντας” τη στήλη στην οποία ανήκει το εικονοστοιχείο από κάτω προς τα πάνω, ανιχνεύουμε τις συντεταγμένες του κέντρου της βάσης του ισοσκελούς καθώς θα είναι το πρώτο μαύρο εικονοστοιχείο που θα συναντήσουμε. Η λογική που χρησιμοποιούμε σε μορφή ψευδοκώδικα είναι η εξής:

```
ΓΙΑ ΚΑΘΕ εικονοστοιχείο στη στήλη της κορυφής από κάτω προς τα πάνω
        {
        ΑΝ (χρώμα_εικονοστοιχείου == μαύρο)
                {
                        ΕΠΙΣΤΡΕΨΕ συντεταγμένες εικονοστοιχείου
                }
        }
```

Γνωρίζοντας πλέον την αρχή και το τέλος του Matrix Barcode, μπορούμε να προχωρήσουμε στο επόμενο βήμα της εύρεσης των ακριβών συντεταγμένων των στοιχείων – τριγώνων.

5.5 Εύρεση συντεταγμένων των στοιχείων του Matrix Barcode σε στιγμιότυπο οθόνης

Ως συντεταγμένες ενός στοιχείου θεωρούμε τις συντεταγμένες ενός οποιουδήποτε εσωτερικού σημείου του στοιχείου κοντά στο κέντρο βάρους του. Ουσιαστικά θέλουμε να πάρουμε μέτρηση των τιμών των χρωμάτων σε ένα σημείο όπου γνωρίζουμε ότι είναι μακριά από τις ακμές του τριγώνου καθώς, λόγω ανάλυσης και ατελειών της εικόνας, κοντά στις ακμές μπορεί να υπάρχει παραμόρφωση της πληροφορίας. Συνεπώς υπολογίζουμε τις συντεταγμένες του κέντρου βάρους

του τριγώνου αγνοώντας οποιεσδήποτε ανακρίβειες υπολογισμού που μπορούν να προκύψουν από το πάχος των γραμμών.

Η εύρεση των συντεταγμένων των στοιχείων πραγματοποιείται σε δύο βήματα. Το πρώτο είναι η εύρεση των συντεταγμένων των κεντρικών τριγώνων της κάθε σειράς. Το δεύτερο βήμα είναι η εύρεση των συντεταγμένων όλων των υπόλοιπων τριγώνων.

5.5.1 Εύρεση συντεταγμένων των κεντρικών στοιχείων κάθε σειράς

Γνωρίζοντας το ανώτατο και κατώτατο σημείο του Matrix Barcode, χρησιμοποιούμε την εξής λογική για την εύρεση των συντεταγμένων των κεντρικών στοιχείων κάθε σειράς:

1. Ξεκινάμε την αναζήτηση από το ανώτατο σημείο του Matrix Barcode προς τα κάτω. Γνωρίζουμε ότι το σημείο έναρξης αντιστοιχεί σε μαύρο εικονοστοιχείο.
2. Αναζητούμε το πρώτο λευκό εικονοστοιχείο. Η εύρεση του πρώτου λευκού εικονοστοιχείου σημαίνει ότι πλέον βρισκόμαστε στο εσωτερικό μέρος του στοιχείου.
3. Αναζητούμε το επόμενο μαύρο εικονοστοιχείο. Όταν το βρούμε, σημαίνει ότι βρήκαμε τη βάση του στοιχείου.
4. Γνωρίζοντας πλέον τις συντεταγμένες της κορυφής του στοιχείου και της βάσης του, υπολογίζουμε το μέσο όρο τους. Αυτές είναι και οι συντεταγμένες του κεντρικού στοιχείου αυτής της γραμμής.
5. Συνεχίζουμε με την ίδια λογική ξεκινώντας από τη βάση του στοιχείου της προηγούμενης σειράς μέχρι να ξεπεράσουμε τις συντεταγμένες του κατώτατου μέρους του Matrix Barcode.

Όσο συμβαίνει αυτή η διαδικασία, υπολογίζουμε ταυτόχρονα το πλήθος των γραμμών και άρα είναι εφικτός και ο υπολογισμός του συνολικού πλήθους των στοιχείων του Matrix Barcode.

Δημιουργούμε λοιπόν έναν πίνακα, όπως στο κεφάλαιο 2, για την αποθήκευση των συντεταγμένων των στοιχείων. Ο δείκτης του κάθε στοιχείου στον πίνακα είναι η σειρά εμφάνισής του στο Matrix Barcode. Έτσι, το πρώτο στοιχείο του πίνακα έχει δείκτη 0 και συντεταγμένες (x_0, y_0), το δεύτερο στοιχείο έχει δείκτη 1 και συντεταγμένες (x_1, y_1) και ούτω καθεξής.

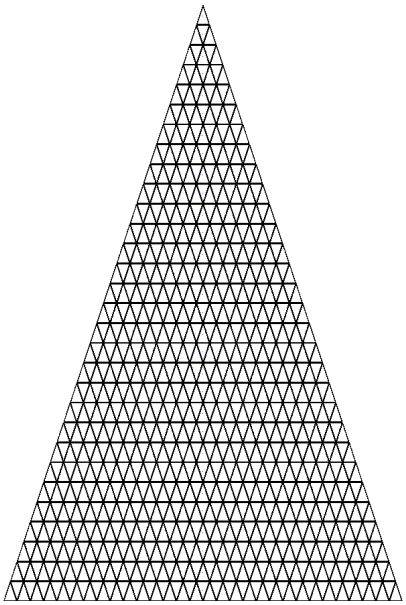
Αυτή τη στιγμή γνωρίζουμε μόνο τις συντεταγμένες των κεντρικών στοιχείων κάθε γραμμής. Έστω ότι το Matrix Barcode έχει N γραμμές (N = 1, 2, 3, …). Τότε κάθε γραμμή περιέχει 2N – 1 στοιχεία. Όπως αναφέρθηκε στην παράγραφο 2.8, το άθροισμα των περιττών αριθμών (2N – 1) είναι N^2. Ακόμη, η σειρά εμφάνισης του κεντρικού στοιχείου στη γραμμή N είναι προφανώς N. Συνεπώς, οι τιμές των δεικτών στον πίνακα που αναζητούμε για το κεντρικό στοιχείο της γραμμής N είναι:

$$(N-1)^2+N=N^2+3N+1$$

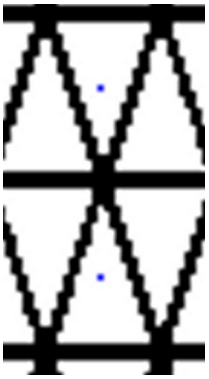
Αντίστοιχα, αν θεωρήσουμε ως πρώτη γραμμή τη γραμμή με δείκτη 0 και ως πρώτο σε σειρά εμφάνισης το στοιχείο με δείκτη 0 προκύπτει ότι το κεντρικό στοιχείο της γραμμής N έχει τιμή δείκτη:

$$N^2+N=N(N+1)$$

Χρησιμοποιώντας τον παραπάνω τύπο, αποθηκεύουμε τις συντεταγμένες των κεντρικών στοιχείων της γραμμής στον πίνακα που δημιουργήσαμε. Για λόγους επαλήθευσης, χρωματίζουμε τα εικονοστοιχεία αυτά με μπλε χρώμα και τα αποθηκεύουμε σε ένα καινούριο αρχείο με όνομα "SavedScreenBlueDotsA.png". Το αποτέλεσμα για το παράδειγμα του κεφαλαίου είναι το εξής:



Σχήμα 36: *Αποτέλεσμα της εύρεσης των κεντρικών στοιχείων της κάθε γραμμής του Matrix Barcode. Στο κέντρο βάρους κάθε τέτοιου τριγώνου υπάρχει ένα μπλε εικονοστοιχείο. Σε αυτό το σημείο θα γίνει η μέτρηση των τιμών χρώματος στην αρχική εικόνα.*



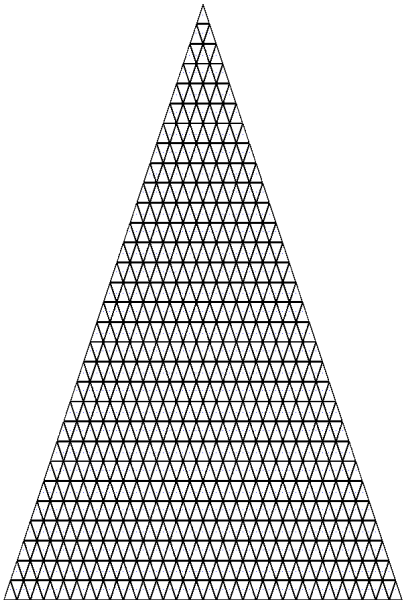
Σχήμα 37: *Λεπτομέρεια από το Σχήμα 36. Οι μπλε βούλες αντιστοιχούν στο κατά προσέγγιση κέντρο βάρος του τριγώνου.*

5.5.2 Εύρεση συντεταγμένων των υπόλοιπων στοιχείων κάθε σειράς

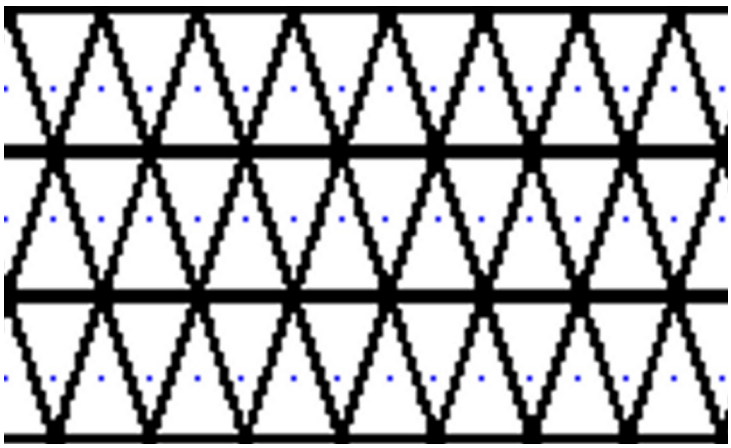
Χρησιμοποιώντας ως σημείο αναφοράς για κάθε γραμμή τις γνωστές, πλέον, συντεταγμένες του κεντρικού της στοιχείου, ακολουθούμε τα εξής βήματα για την εύρεση των συντεταγμένων όλων των υπόλοιπων τριγώνων:

1. Ξεκινάμε την αναζήτηση από το κεντρικό στοιχείο κάθε γραμμής. Πραγματοποιούμε πρώτα την αναζήτηση προς αριστερά.
2. Κάθε φορά που βρίσκουμε ένα μοτίβο μαύρων, άσπρων και μετά μαύρων εικονοστοιχείων αποθηκεύουμε τις συντεταγμένες των μαύρων εικονοστοιχείων και υπολογίζουμε το μέσο όρο τους. Αν το κεντρικό στοιχείο έχει δείκτη στον πίνακα k τότε οι συντεταγμένες που υπολογίσαμε αντιστοιχούν στο στοιχείο k-1 για το πρώτο στοιχείο στα αριστερά του.
3. Συνεχίζουμε τη διαδικασία προς τα αριστερά, μετρώντας πόσα στοιχεία βρήκαμε και σταματώντας όταν υπερβούμε τον γνωστό αριθμό στοιχείων που υπάρχουν προς τα αριστερά. Ταυτόχρονα αποθηκεύουμε τις συντεταγμένες τους στον πίνακα.
4. Μόλις ολοκληρωθεί η εύρεση των συντεταγμένων προς τα αριστερά, ακολουθούμε την ίδια διαδικασία προς τα δεξιά.
5. Με την ολοκλήρωση της εύρεσης όλων των συντεταγμένων των στοιχείων σε αυτή τη γραμμή προχωράμε στην επόμενη γραμμή.
6. Όταν φτάσουμε στην τελευταία γραμμή, ολοκληρώνουμε τη διαδικασία.

Για λόγους επαλήθευσης, χρωματίζουμε ξανά τα εικονοστοιχεία που βρήκαμε με μπλε χρώμα και τα αποθηκεύουμε σε ένα καινούριο αρχείο με όνομα “SavedScreenBlueDotsB.png”. Το αποτέλεσμα για το παράδειγμα του κεφαλαίου είναι το εξής:



Σχήμα 38: *Το οπτικό αποτέλεσμα της εύρεσης των συντεταγμένων στο στιγμιότυπο οθόνης του παραδείγματος του κεφαλαίου. Κάθε στοιχείο του Matrix Barcode έχει ένα μπλε εικονοστοιχείο μέσα του που δείχνει το σημείο στο οποίο θα γίνει η μέτρηση των τιμών του χρώματος στην αρχική εικόνα.*



Σχήμα 39: *Λεπτομέρεια από την εικόνα του Σχήματος 38. Όλα τα τρίγωνα περιέχουν ένα μπλε εικονοστοιχείο μέσα τους στις συντεταγμένες που θα γίνει η μέτρηση των τιμών του χρώματος στην αρχική εικόνα.*

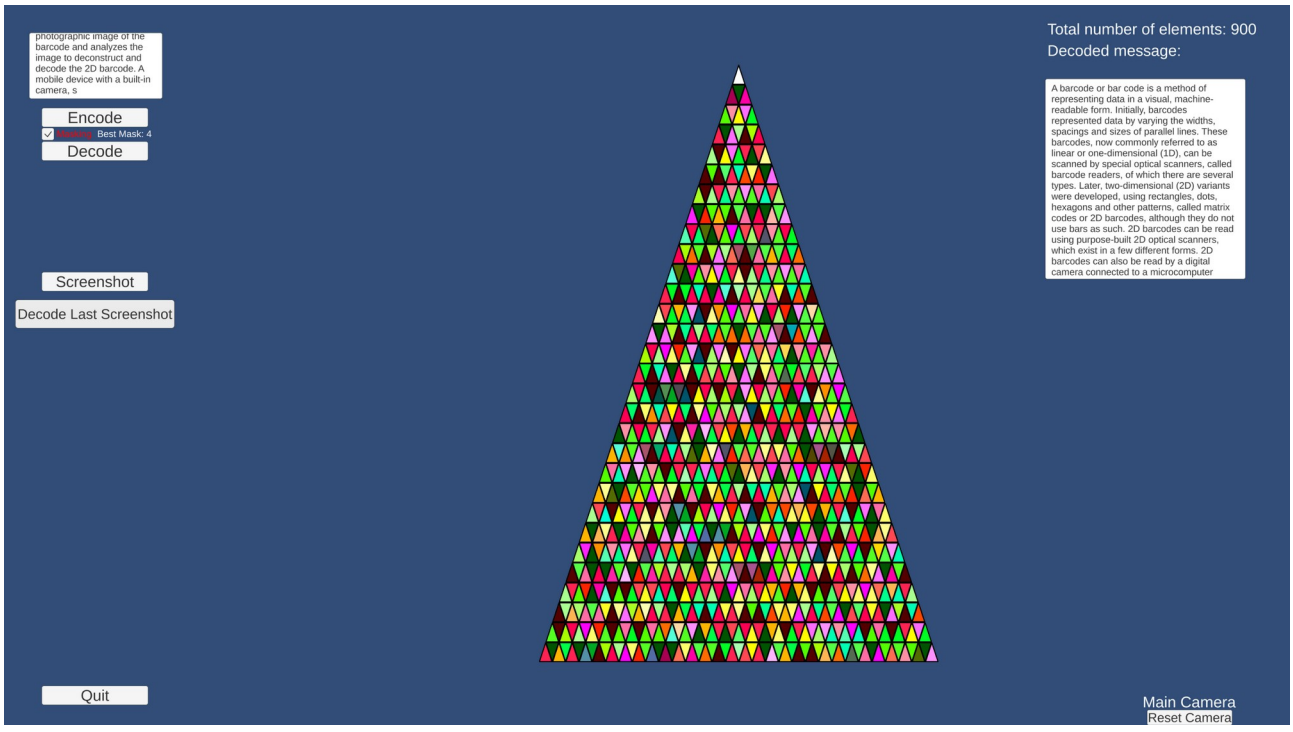
Όπως φαίνεται και από τα Σχήματα 38 και 39, η διαδικασία έχει ολοκληρωθεί και πλέον είναι γνωστές οι συντεταγμένες όλων των στοιχείων του Matrix Barcode και είναι αποθηκευμένες σειριακά σε έναν πίνακα.

5.6 Αποκωδικοποίηση των χρωμάτων του Matrix Barcode από στιγμιότυπο οθόνης

Γνωρίζοντας πλέον τις συντεταγμένες των στοιχείων στην εικόνα, επιστρέφουμε στην αρχική έγχρωμη έκδοσή της και “διαβάζουμε” τις RGB τιμές των χρωμάτων. Στηριζόμενοι στις συναρτήσεις της αποκωδικοποίησης που περιγράφηκαν στο κεφάλαιο 3 της αποκωδικοποίησης, χρησιμοποιούμε την εξής λογική για την αποκωδικοποίηση του μηνύματος:

1. Αρχικά βρίσκουμε την ταυτότητα της μάσκας συγκρίνοντας το χρώμα του πρώτου στοιχείου με τα τέσσερα διαθέσιμα χρώματα μασκών.
2. Για κάθε στοιχείο (εκτός του πρώτου) βρίσκουμε το χρώμα του από την έγχρωμη εικόνα και του αφαιρούμε τη μάσκα που βρήκαμε με βάση τους κανόνες της.
3. Αποκωδικοποιούμε το χρώμα που προκύπτει από το (2) και το αντιστοιχούμε στον χαρακτήρα ASCII 7-bit.
4. Προσθέτουμε τον χαρακτήρα ASCII 7-bit στο αποκωδικοποιημένο μήνυμα και συνεχίζουμε μέχρι να αποκωδικοποιηθούν όλα τα στοιχεία.

Στο τέλος της αποκωδικοποίησης, το μήνυμα εμφανίζεται στο πεδίο κειμένου στη δεξιά πλευρά της διεπαφής χρήστη. Για την έναρξη της αποκωδικοποίησης ο χρήστης πρέπει να πατήσει το κουμπί “Decode Last Screenshot”. Έτσι για το παράδειγμα του κεφαλαίου παίρνουμε το εξής αποτέλεσμα:



Σχήμα 40: Αποτέλεσμα αποκωδικοποίησης από το στιγμιότυπο οθόνης που έβγαλε ο χρήστης. Το αποκωδικοποιημένο μήνυμα φαίνεται στη δεξιά πλευρά της διεπαφής χρήστη.

Η επιτυχής ολοκλήρωση της αποκωδικοποίησης από ένα στιγμιότυπο οθόνης μας οδηγεί στο συμπέρασμα ότι το Matrix Barcode της παρούσης διπλωματικής πληροί τις προδιαγραφές ενός ψηφιακού Matrix Barcode (που εμφανίζεται μόνο σε οθόνη) και είναι δυνατή η περαιτέρω εξέλιξή του για να ανταποκρίνεται ορθά και σε πραγματικές συνθήκες (Matrix Barcode σε εκτυπώσιμη μορφή).

Κεφάλαιο 6. Συμπεράσματα και μελλοντικές βελτιώσεις

Στο κεφάλαιο αυτό, στην παράγραφο 6.1 παρουσιάζουμε τα συμπεράσματα στα οποία καταλήξαμε μετά την ανάπτυξη και δοκιμή του Matrix Barcode της παρούσης διπλωματικής. Στην παράγραφο 6.2 παρουσιάζουμε κάποιες ιδέες για πιθανές βελτιώσεις της απόδοσης και αξιοπιστίας του Matrix Barcode.

6.1 Συμπεράσματα

Η δημιουργία ενός Matrix Barcode από την αρχή είναι μια απαιτητική διαδικασία όπου πρέπει να δοθεί έμφαση σε κάθε λεπτομέρεια με βάση τις ανάγκες του. Το Matrix Barcode της παρούσης διπλωματικής, στη μορφή που βρίσκεται, μπορεί να λειτουργήσει αξιόπιστα όσο μεταδίδεται με ψηφιακά μέσα τα οποία δεν εισάγουν αξιοσημείωτη παραμόρφωση.

Η ύπαρξη 128 = 2^7 χρωμάτων το καθιστά επιρρεπές σε σφάλματα μετρήσεων και παραμορφώσεις καθώς το κομμάτι της ανάγνωσης της τιμής ενός χρώματος κατά τη διάρκεια της αποκωδικοποίησης απαιτεί πολύ περισσότερη ακρίβεια από ένα αντίστοιχο ασπρόμαυρο Matrix Barcode. Από την άλλη, η συμπύκνωση της πληροφορίας ανά μονάδα χώρου είναι της τάξης του:

$$\frac{\textit{Εύρος πληροφοριών ανά 7 bit στοιχείο του Matrix Barcode}}{\textit{Εύρος πληροφοριών ανά 1 bit στοιχείο του Matrix Barcode}} = \frac{2^7}{2^1} = 2^6 = 64$$

Ο αριθμός που προκύπτει σημαίνει πως το Matrix Barcode της παρούσης διπλωματικής χρειάζεται 64 φορές λιγότερο χώρο για να κωδικοποιήσει ένα μήνυμα σε σχέση με ένα ασπρόμαυρο Matrix Barcode.

Όσον αφορά στο τριγωνικό σχήμα που επιλέχθηκε, μας επιτρέπει να αποφύγουμε την προσθήκη κάποιου μοτίβου ευθυγράμμισης που περιγράψαμε στην παράγραφο 1.2.2. Το σχήμα κάθε στοιχείου (αλλά και ολόκληρο το Matrix Barcode) εμπεριέχει το σωστό προσανατολισμό του ενώ τα περιγράμματα των στοιχείων μπορούν να χρησιμοποιηθούν ως βοήθημα στην ευθυγράμμιση σε περίπτωση που αυτό χρειαστεί. Αυτό σημαίνει ότι γίνεται εξοικονόμηση χώρου στο Matrix Barcode καθώς υπάρχει περισσότερος χώρος για την πληροφορία.

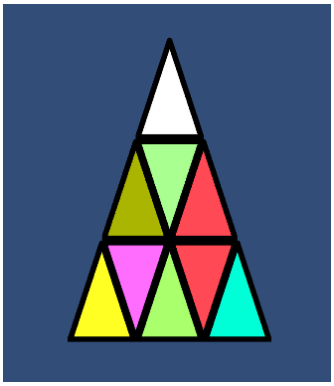
Ακόμη, η ύπαρξη τόσων χρωμάτων καθώς και το ασυνήθιστο σχήμα του, σε σχέση με τα υπόλοιπα Matrix Barcode, προσδίδει ένα σαφώς αναβαθμισμένο αισθητικό αποτέλεσμα. Καθώς η χρήση του πιο δημοφιλούς Matrix Barcode (QR Code) συνεχίζει να αυξάνεται και η ανάγκη των επιχειρήσεων να ξεχωρίζουν από τους ανταγωνιστές τους παραμένει αναλλοίωτη, είναι ασφαλές να υποθέσουμε ότι η ανάπτυξη πρωτότυπων Matrix Barcode, όπως αυτό της παρούσας διπλωματικής ή το JAB Code, θα συνεχιστεί και ίσως αυξηθεί.

6.2 Μελλοντικές βελτιώσεις

Παρόλο που το Matrix Barcode που κατασκευάσαμε είναι λειτουργικό σε ιδανικές συνθήκες, υπάρχουν τρόποι βελτίωσης της απόδοσής του ώστε να είναι εφικτή η αποκωδικοποίησή του σε πραγματικές συνθήκες. Ως πραγματικές συνθήκες εννοούμε το “σκανάρισμα” του Matrix Barcode

από μια κάμερα κινητού σε έναν τυχαίο προσανατολισμό με μη ιδανικό φωτισμό. Ενδεικτικά, αναφέρουμε μερικά χαρακτηριστικά τα οποία θα βελτιώσουν την απόδοση και την αξιοπιστία του Matrix Barcode.

1. Ίσως η πιο σημαντική προσθήκη στο Matrix Barcode είναι η εισαγωγή κάποιας τεχνικής διόρθωσης σφάλματος. Οι αλγόριθμοι διόρθωσης σφάλματος είναι αναπόσπαστο στοιχείο των Matrix Barcode και μπορούν να διορθώσουν τυχαία σφάλματα (random errors) ή σφάλματα ριπής (burst errors) χρησιμοποιώντας μικρό αναλογικά χώρο στο Matrix Barcode σε σχέση με αυτόν που καταλαμβάνει η πληροφορία. Το error correction κρίνεται απαραίτητο ειδικά σε περιπτώσεις όπου το κωδικοποιημένο μήνυμα είναι, παραδείγματος χάρη, μια ιστοσελίδα στο διαδίκτυο ή ένα μοναδικό αναγνωριστικό (όπως ενα IBAN μιας τράπεζας), όπου το σφάλμα ακόμα και σε έναν χαρακτήρα μπορεί να καταστρέψει όλο το υπόλοιπο μήνυμα.
2. Μια πιθανή προσθήκη θα μπορούσε να είναι ένα μοτίβο εύρεσης. Το μοτίβο εύρεσης, ουσιαστικά ανιχνεύει αν υπάρχει ή όχι το συγκεκριμένο Matrix Barcode στην φωτογραφία και μετά αποκωδικοποιεί το μήνυμα. Στην περίπτωση του Matrix Barcode της παρούσης διπλωματικής, η εφαρμογή επιχειρεί να αποκωδικοποιήσει κάθε στιγμιότυπο οθόνης που της δίνεται χωρίς να μπορεί να αναγνωρίσει αν υπάρχει κάτι προς αποκωδικοποίηση.
3. Βασική προϋπόθεση της λειτουργίας των Matrix Barcode είναι επίσης και η χρονική τους απόκριση, δηλαδή το πόσο γρήγορα γίνεται η αποκωδικοποίηση ενός μηνύματος. Η αποκωδικοποίηση από στιγμιότυπο οθόνης γίνεται περίπου σε 1-4 δευτερόλεπτα και εξαρτάται από το μέγεθος του Matrix Barcode, αλλά με μερικές βελτιστοποιήσεις στον κώδικα και στη λογική, ο χρόνος μπορεί να μειωθεί σημαντικά.
4. Τέλος, μια λειτουργία που μπορεί να προστεθεί είναι η ανίχνευση του ορθού προσανατολισμού του Matrix Barcode χρησιμοποιώντας τεχνικές επεξεργασίας εικόνας και αξιοποιώντας τη φύση του ισοσκελούς τριγώνου που μπορεί να δηλώσει με σαφή τρόπο τον ορθό προσανατολισμό του.



Σχήμα 41: *Το Matrix Barcode της παρούσης διπλωματικής. Το κωδικοποιημένο μήνυμα περιλαμβάνει το όνομα που του αποδόθηκε.*

Βιβλιογραφία

[1] Barcode, Wikipedia, 2023, https://en.wikipedia.org/wiki/Barcode
[2] QR Code, Wikipedia, 2023, https://en.wikipedia.org/wiki/QR_code
[3] JAB Code, Wikipedia, 2023, https://en.wikipedia.org/wiki/JAB_Code
[4] JAB Code, Fraunhofer SIT, 2023, https://jabcode.org/
[5] High Capacity Color Barcode, Wikipedia, 2023, https://en.wikipedia.org/wiki/High_Capacity_Color_Barcode
[6] Cronto Visual Cryptogram, One Span, 2023, https://www.onespan.com/products/transaction-signing/cronto
[7] QR Code Tutorial, Thonky, 2023, https://www.thonky.com/qr-code-tutorial/
[8] Character Encoding, Wikipedia, 2023, https://en.wikipedia.org/wiki/Character_encoding
[9] RGB color model, Wikipedia, 2023, https://en.wikipedia.org/wiki/RGB_color_model
[10] 8-bit color, Wikipedia, 2023, https://en.wikipedia.org/wiki/8-bit_color