

**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

Περιεχόμενα

Ευχαριστίες.....	3
Abstract.....	5
Περίληψη.....	6
Κεφάλαιο 1. Εισαγωγή.....	8
1.1 Έγχρωμα υπάρχοντα Matrix Barcode.....	8
1.1.1 QR code.....	8
1.1.2 JAB code.....	9
1.1.3 High Capacity Color Barcode.....	9
1.1.4 Cronto Visual Cryptogram.....	10
1.2 Βασικά χαρακτηριστικά των Matrix Barcode.....	11
1.2.1 Finder Pattern (μοτίβο εύρεσης).....	11
1.2.2 Alignment Pattern (μοτίβο ευθυγράμμισης).....	11
1.2.3 Timing Pattern (μοτίβο χρονισμού).....	12
1.2.4 Error Correction Code (κώδικας διόρθωσης σφάλματος).....	13
1.2.5 Masking.....	13
Κεφάλαιο 2. Κωδικοποίηση.....	15
2.1 Κωδικοποίηση χαρακτήρα (Character Encoding).....	15
2.2 Κωδικοποίηση σε Matrix Barcode.....	16
2.3 Μοντέλο RGB απεικόνισης χρωμάτων.....	16
2.4 Υπολογισμός των τιμών χρώματος στη Unity.....	17
2.4 Κωδικοποίηση χαρακτήρα σε Matrix Barcode στη Unity.....	18
2.5 Δημιουργία και τοποθέτηση του στοιχείου του Matrix Barcode στη Unity....	18
Κεφάλαιο 3. Masking (τεχνικές απόκρυψης).....	19
3.1 α.....	19
3.1.1 α.....	19
Κεφάλαιο 4. Αποκωδικοποίηση.....	20
4.1 α.....	20
4.1.1 α.....	20
Κεφάλαιο 5. Αποκωδικοποίηση από Screenshot.....	21
5.1 α.....	21
5.1.1 α.....	21
Κεφάλαιο 6. Συμπεράσματα και πιθανές βελτιώσεις.....	22
6.1 α.....	22
6.1.1 α.....	22

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

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- πρόκειται για 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 της 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 της 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

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.

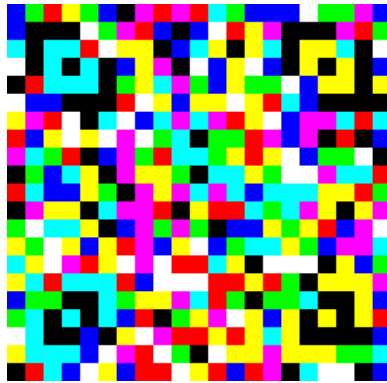


Σχήμα 1: QR code για το URL (διεύθυνση ιστοσελίδας) της αγγλόφωνης Wikipedia σε έκδοση κινητού

1.1.2 JAB code

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.

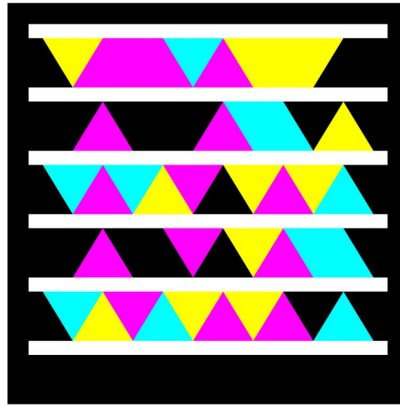
Σε σύγκριση με άλλα συμβατικά ασπρόμαυρα Matrix Barcodes, το JAB code έχει τη δυνατότητα να περιέχει στην ίδια περιοχή του κώδικα δύο ή τρεις φορές περισσότερες πληροφορίες για χρήση τεσσάρων ή οχτώ χρωμάτων αντίστοιχα. Με βάση αυτή τη δυνατότητά του, είναι ακόμα πιο εφικτή η αποθήκευση ενός ολόκληρου 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

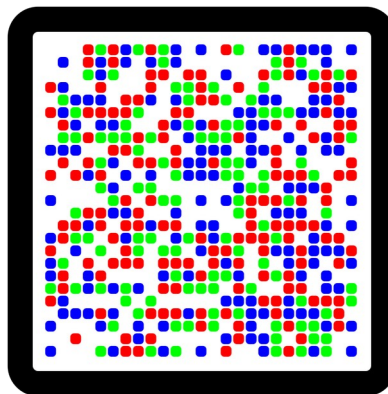
Το High Capacity Color Barcode αποτελεί μια τεχνολογία που αναπτύχθηκε από τη Microsoft για την κωδικοποίηση δεδομένων χρησιμοποιώντας συστάδες έγχρωμων τριγώνων αντί των συμβατικών τετραγώνων. Η πυκνότητα πληροφορίας είναι μεγαλύτερη σε σύγκριση με τα μονόχρωμα Matrix Barcodes καθώς χρησιμοποιείται 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

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.



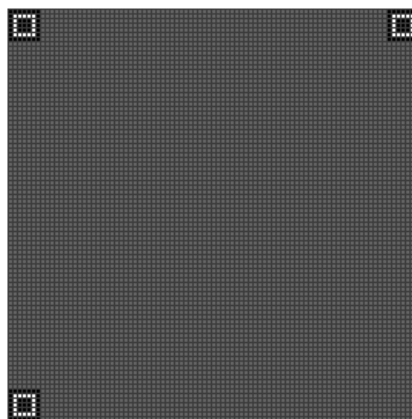
Σχήμα 4: Μία απεικόνιση ενός *photoTAN code* με τυχαία δεδομένα

1.2 Βασικά χαρακτηριστικά των Matrix Barcode

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)

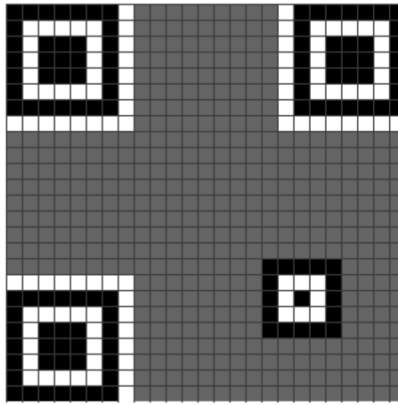
Το Finder Pattern ή αλλιώς μοτίβο εύρεσης αποτελεί ίσως το πιο βασικό χαρακτηριστικό των Matrix Barcode. Ανήκει στην κατηγορία των function patterns (λειτουργικά μοτίβα) που είναι ουσιαστικά συγκεκριμένα σχήματα τα οποία εμφανίζονται σε προκαθορισμένα μέρη του κώδικα που δεν περιέχουν δεδομένα. Το Finder 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)

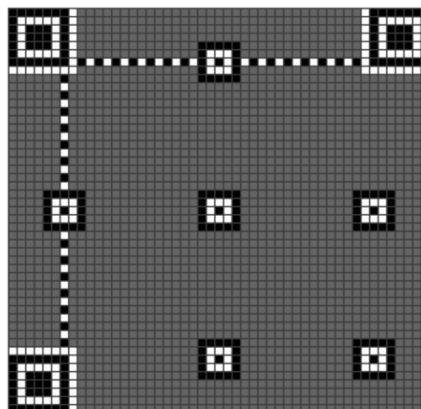
Το Alignment pattern ή μοτίβο ευθυγράμμισης είναι ένα ακόμη βασικό χαρακτηριστικό των Matrix Barcodes. Ανήκει κι αυτό στην κατηγορία των function patterns και η λειτουργία του είναι να βοηθάει στην ανίχνευση του ορθού προσανατολισμού από μια συσκευή απεικόνισης. Ουσιαστικά, η λειτουργία του είναι να 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.



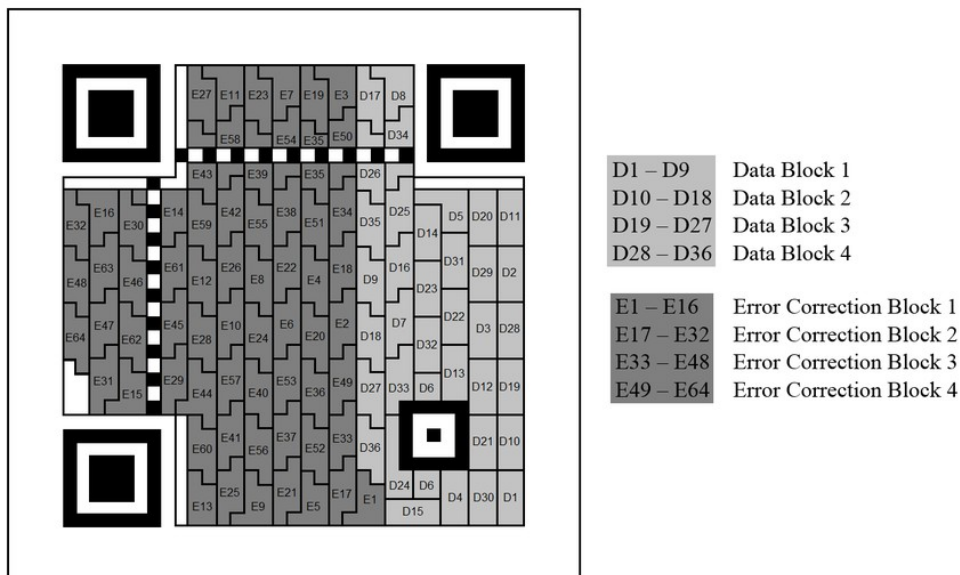
Σχήμα 7: Timing patterns στον οριζόντιο και κατακόρυφο άξονα μαζί με τα τρία Finder Pattern και έξι alignment pattern ενός QR Code

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)

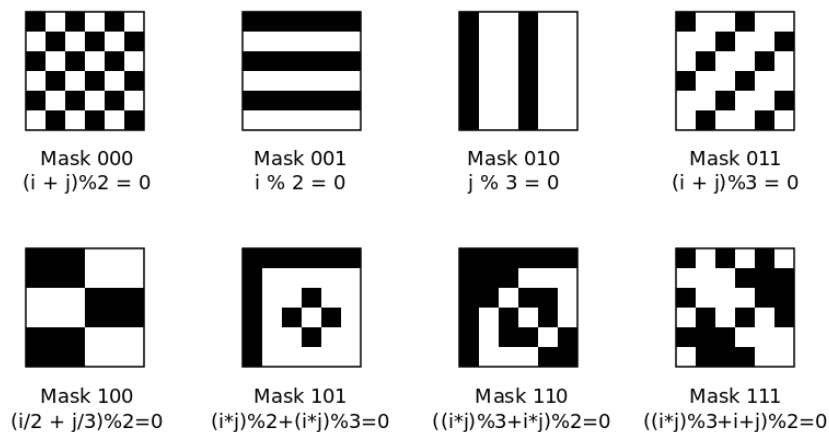
1.2.5 Masking

Το Masking είναι μια τεχνική επικάλυψης και απόκρυψης της πληροφορίας. Χρησιμοποιείται σε πολλά Matrix Barcode για την αποφυγή εμφάνισης ίδιας ή παρόμοιας πληροφορίας συνεχόμενα σε ένα Matrix Barcode. Χρησιμοποιώντας μια τεχνική masking, η οποία αλλάζει τη μορφή εμφάνισης ακολουθώντας συγκεκριμένους κανόνες, είναι εφικτή η δημιουργία ενός κώδικα ο οποίος θα αποκρύπτει την πληροφορία και θα αποκωδικοποιείται πιο εύκολα.

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.

Στη μεγαλύτερη πλειοψηφία των Matrix Barcode, υπάρχουν κανάλια (presets) masking. Μετά την κωδικοποίηση, γίνεται αξιολόγηση του κάθε καναλιού masking και χρησιμοποιείται το βέλτιστο για την εκάστοτε περίπτωση. Αφού επιλεγθεί το κανάλι, τυπώνεται στον κώδικα με τον κατάλληλο τρόπο έτσι ώστε να είναι δυνατή 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.

Κεφάλαιο 2. Κωδικοποίηση

Στο κεφάλαιο της κωδικοποίησης θα εξετάσουμε

2.2 τι είναι κωδικοποίηση σε matrix barcodes + bitwise operations συνήθως αλλά εδώ όχι είμαστε με χρωμάτα

κσεkina des document

ποιο είναι το plano. De 8a exei module error correction de 8a exei χρονισμού κλπ

Stoixeio trigwnaki

pws xrisimopoiéis to trigwnaki, 8eseis

odigies programmatos

pes gia maska oti to prwto trigwno to desmepsame

deikse screen etoimo encoded prama

seiriaka ola prwtos xaraktiras prwto trigwnaki κλπ

2.1 Κωδικοποίηση χαρακτήρα (Character Encoding)

Η κωδικοποίηση, σαν όρος, είναι η διαδικασία μετατροπής πληροφοριών σε μια διαφορετική μορφή η οποία είναι κατάλληλη για αποθήκευση ή μετάδοση πληροφοριών. 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 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 αναπτύχθηκε, επιλέχθηκε η 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

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 και θα πρέπει να γίνει με τέτοιο τρόπο ώστε να είναι εφικτή και η αποκωδικοποίηση. 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 είναι εφικτός ο ορισμός μια μετατροπής από τη binary μορφή τους σε μια μοναδική τους απεικόνιση στο Matrix Barcode. Όπως αναφέρθηκε προηγουμένως, τα περισσότερα 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 αναπτύχθηκε, 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.

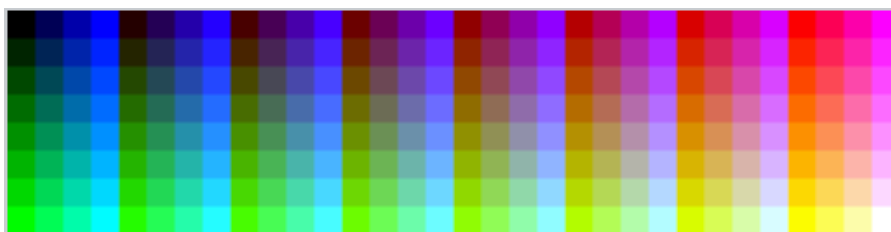
Όπως είναι γνωστό, ένα από τα πιο διαδεδομένα μοντέλα περιγραφής χρωμάτων είναι το RGB (Red, Green, Blue) σύμφωνα με το οποίο είναι εφικτή η δημιουργία οποιουδήποτε χρώματος στο ανθρώπινο ορατό φάσμα. Το μοντέλο αυτό χρησιμοποιεί τα χρώματα κόκκινο, πράσινο και μπλε για την περιγραφή ενός χρώματος. Οι τιμές κάθε χρώματος μπορούν να πάρουν από την τιμή μηδέν μέχρι και ένα μέγιστο το οποίο ορίζεται από το εκάστοτε λογισμικό που χρησιμοποιείται. Στην περίπτωση της 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.

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 στη Unity

Το κάθε χρώμα περιγράφεται στη 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). Επειδή χρειάζεται να χωρίσουμε το κλειστό διάστημα [0, 1] σε 4 μέρη, καταλήγουμε (κατά προσέγγιση) στις float τιμές

$$\frac{0}{3}=0, \frac{1}{3}\approx 0.33, \frac{2}{3}\approx 0.67 \text{ και } \frac{3}{3}=1.$$

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:

	K = Κόκκινο	Π = Πράσινο	M = Μπλε
0	$K_0 = 0$	$\Pi_0 = 0$	$M_0 = 0$
1	$K_1 = 0.33$	$\Pi_1 = 0.14$	$M_1 = 0.33$
2	$K_2 = 0.67$	$\Pi_2 = 0.28$	$M_2 = 0.67$
3	$K_3 = 1$	$\Pi_3 = 0.43$	$M_3 = 1$
4	-----	$\Pi_4 = 0.57$	-----
5	-----	$\Pi_5 = 0.71$	-----
6	-----	$\Pi_6 = 0.86$	-----
7	-----	$\Pi_7 = 1$	-----

Πίνακας 1: Πίνακας Αποχρώσεων

2.5 Κωδικοποίηση χαρακτήρα σε Matrix Barcode στη 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.

- Μέσω μιας εντολής στη Unity, γίνεται η μετατροπή από τον χαρακτήρα ASCII στη μοναδική φυσική τιμή του (στο δυαδικό ανάπτυγμα). Δηλαδή

$$\Psi\eta\phi\acute{\iota}\sigma\ \text{ASCII} \leftrightarrow \text{φυσικό}\ b \in [0,127] \leftrightarrow b = \sum_{i=0}^6 b_i 2^i = (b_6, b_5, b_4, b_3, b_2, b_1, b_0)$$

- 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), 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 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 μοναδικό χρώμα.

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

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

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

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

Έτσι στο παραπάνω παράδειγμα έχουμε:

$$\mathbf{1\ 1\ 1\ 1\ 0\ 1\ 0} = 122$$

Bit	7	6	5	4	3	2	1	0
Data	-	1	1	1	1	0	1	0

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

$$\text{Blue} \rightarrow 122 \bmod 2^2 \div 2^0 = 122 \bmod 4 = 2 \rightarrow \text{blue color} = 0.6667$$

$$\text{Green} \rightarrow 122 \bmod 2^5 \div 2^2 = 26 \div 4 = 6 \rightarrow \text{green color} = 0.857$$

$$\text{Red} \rightarrow 122 \bmod 2^7 \div 2^5 = 122 \div 32 = 3 \rightarrow \text{red color} = 1$$

2.6 Εγκυρότητα μηνύματος

Έχοντας θέσει τους βασικούς κανόνες της κωδικοποίησης που περιγράφουν τις αντιστοιχίες κάθε ASCII 7-bit χαρακτήρα σε μοναδικά χρώματα, χρειάζεται να γίνει έλεγχος του μηνύματος προς κωδικοποίηση για 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 την εγκυρότητα του μηνύματος ως προς το μέγεθός του, not_null είναι η boolean μεταβλητή που παίρνει την τιμή αληθής όταν το μήνυμα δεν είναι κενό, το size_of_message είναι το πλήθος των χαρακτήρων που περιέχει το μήνυμα και το && η λογική πράξη ΚΑΙ.

Για τον έλεγχο του περιεχομένου του μηνύματος και το αν αυτό περιέχει μόνο ASCII 7-bit χαρακτήρες, γίνεται μετατροπή του μηνύματος σε ASCII bytes και στη συνέχεια η αντίστροφη διαδικασία για να παραχθεί ξανά το μήνυμα. Αν το μήνυμα που παρήχθη είναι το ίδιο με το αρχικό, τότε το μήνυμα περιέχει μόνο ASCII χαρακτήρες. Σε οποιαδήποτε άλλη περίπτωση το μήνυμα θα περιέχει μη έγκυρους χαρακτήρες. Το αποτέλεσμα αυτού του ελέγχου αποθηκεύεται σε μια boolean μεταβλητή η οποία ονομάζεται is_ASCII και παίρνει τιμή αληθής όταν το μήνυμα περιέχει μόνο ASCII χαρακτήρες.

Τελικά το μήνυμα περνάει επιτυχώς τον έλεγχο εγκυρότητας αν η παρακάτω λογική πράξη παράγει την τιμή αληθής:

```
string_is_valid = is_ASCII && valid_size
```

όπου string_is_valid είναι η τελική boolean που περιγράφει αν τελικά το μήνυμα προς κωδικοποίηση είναι έγκυρο ή όχι.

2.7 Δημιουργία και τοποθέτηση του στοιχείου του Matrix Barcode στη 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.

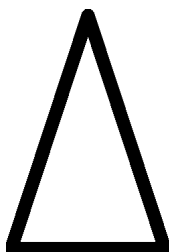


Figure 14: Το στοιχείο του Matrix Barcode

Για κάθε χαρακτήρα που υπάρχει στο μήνυμα, παράγουμε ένα τρίγωνο. Χρωματίζουμε το εσωτερικό του τριγώνου με βάση τον χαρακτήρα που κωδικοποιείται κάθε φορά και το τοποθετούμε στην κατάλληλη θέση έτσι ώστε να σχηματιστεί ένα μεγαλύτερο όμοιο ισοσκελές τρίγωνο.

Η ορθή τοποθέτηση του τριγώνου στην κατάλληλη θέση προϋποθέτει τη δημιουργία ενός προκατασκευασμένου αντικειμένου στη Unity (prefab GameObject). Το πλεονέκτημα των prefab αντικειμένων είναι ότι μπορούμε μέσω μιας εντολής να τα αντιγράψουμε και να τα εμφανίσουμε στον χρήστη. Ένα Game Object της Unity μπορεί να έχει πολλά διαφορετικά Component (συστατικά), αλλά τα πιο σημαντικά που θα μας απασχολήσουν είναι το Component μετασχηματισμού (Transform), το Component Sprite Renderer και το Script Component. Το Transform περιέχει τις καρτεσιανές συντεταγμένες του και την περιστροφή του γύρω από τους άξονες στον εικονικό τρισδιάστατο χώρο του προγράμματος. Εφόσον το Matrix Barcode είναι μια δισδιάστατη οντότητα, χρησιμοποιούμε το επίπεδο των αξόνων (x, y) και ως παρατηρητές βρισκόμαστε στον θετικό ημιάξονα του z. Το Sprite Renderer δίνει τη δυνατότητα απεικόνισης δισδιάστατων γραφικών, και κατ' επέκταση χρωμάτων, πάνω σε ένα αντικείμενο. Το Script επιτρέπει την προσθήκη και συγγραφή κώδικα που αφορά το αντικείμενο αυτό.

Ως αρχική θέση, θεωρούμε τις συντεταγμένες στις οποίες τοποθετείται το πρώτο στοιχείο του Matrix Barcode που βρίσκεται στο ανώτατο μέρος του μεγάλου τριγώνου που θα σχηματιστεί. 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 μέγεθος του αντικειμένου χρησιμοποιώντας μια εντολή στη Unity, συνθέτουμε το Τριγωνικό Matrix Barcode με αναφορά στις συντεταγμένες του πρώτου στοιχείου. Έτσι, αν θεωρήσουμε (x, y) τις συντεταγμένες του στοιχείου κ για κ = 0, 1, 2,... (υπενθυμίζουμε ότι στον προγραμματισμό, το πρώτο στοιχείο ενός πίνακα Α σημειώνεται ως Α[0]) έχουμε:

$$x = \text{αρχική_θέση} \cdot x - \frac{\text{αριθμός_γραμμής} * \text{βάση_ισοσκελούς} + \text{βάση_ισοσκελούς} * v}{2}$$

και

$$y = \text{αρχική_θέση} \cdot y - \text{αριθμός_γραμμής} * \text{ύψος_ισοσκελούς}$$

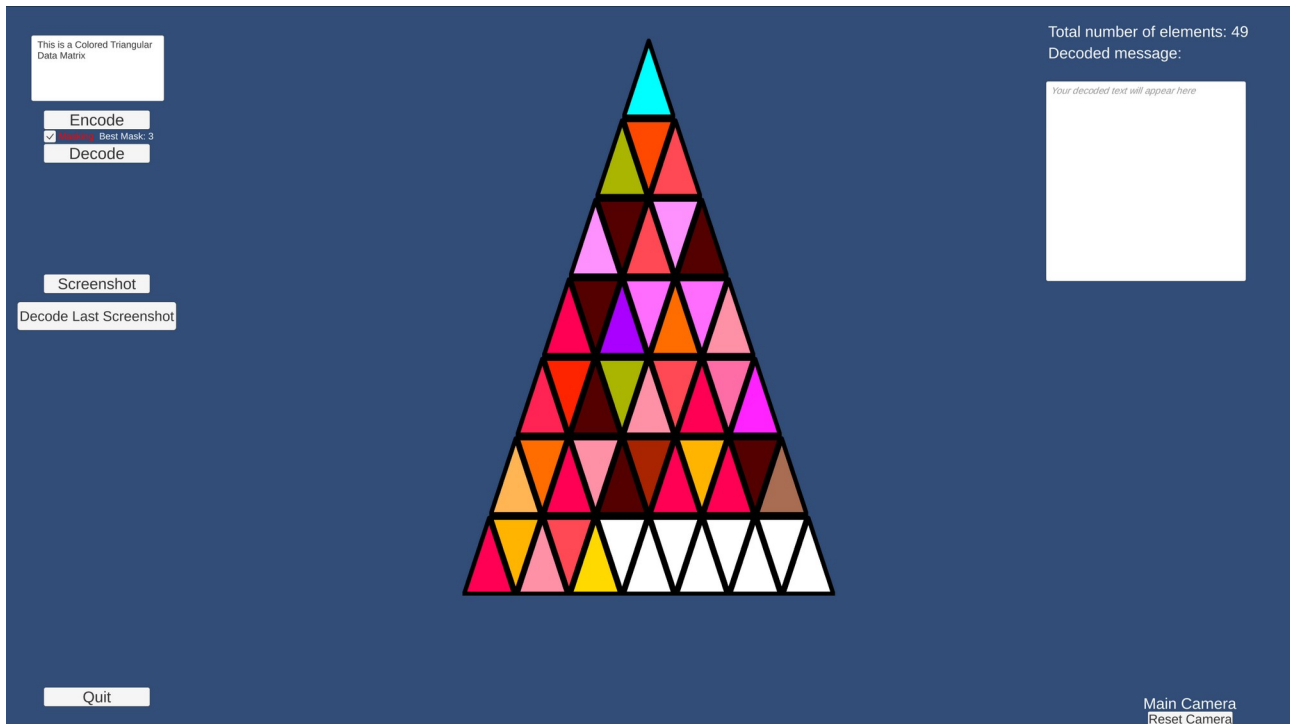
όπου ο αριθμός_γραμμής παίρνει τιμές 0,1,2... για την πρώτη, δεύτερη, τρίτη γραμμή, κλπ.

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. Στο πρώτο τρίγωνο, το οποίο χρησιμοποιείται για masking, αναθέτουμε το κυανό χρώμα και συνεχίζουμε στο δεύτερο τρίγωνο με την κωδικοποίηση του πρώτου χαρακτήρα του μηνύματος. Αν το πλήθος των στοιχείων δεν είναι αρκετό για να ολοκληρώσουν πλήρως τη γραμμή ενός τριγώνου, παράγουμε κενά τρίγωνα μέχρι να συμπληρωθεί ορθά 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 στη 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 15: 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:

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

Κεφάλαιο 3. Masking (τεχνικές απόκρυψης)

Στο κεφάλαιο αυτό,

3.1 α

Κατά τη συγγραφή αυτού του εγγράφου

3.1.1 α

Το

Κεφάλαιο 4. Αποκωδικοποίηση

Στο κεφάλαιο αυτό,

4.1 α

Κατά τη συγγραφή αυτού του εγγράφου

4.1.1 α

Το

Κεφάλαιο 5. Αποκωδικοποίηση από Screenshot

Στο κεφάλαιο αυτό,

5.1 α

Κατά τη συγγραφή αυτού του εγγράφου

5.1.1 α

Το

Κεφάλαιο 6. Συμπεράσματα και πιθανές βελτιώσεις

Στο κεφάλαιο αυτό,

6.1 α

Κατά τη συγγραφή αυτού του εγγράφου

6.1.1 α

Το