

ΠΩΣ ΝΑ ΜΕΤΑΦΡΑΣΕΤΕ ΠΑΙΧΝΙΔΙΑ ΤΟΥ AMSTRAD CPC ΣΤΑ ΕΛΛΗΝΙΚΑ

v.1.0



ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΠΕΡΙΕΧΟΜΕΝΑ	2
2.	ΕΙΣΑΓΩΓΗ	3
3.	ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ ΚΑΙ ΠΕΡΙΕΧΟΜΕΝΑ	4
4.	ΓΡΑΜΜΑΤΟΣΕΙΡΑ ΚΑΙ ΚΩΔΙΚΟΠΟΙΗΣΗ	5
5.	ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΝΟΥ	13
6.	ΑΛΛΑΓΗ ΓΡΑΦΙΚΩΝ ΚΑΙ ΕΙΚΟΝΩΝ	18
7.	ΔΙΑΤΑΞΗ ΜΝΗΜΗΣ, ΡΟΥΤΙΝΕΣ ΣΥΣΤΗΜΑΤΟΣ, BASIC	21
8.	ΔΙΑΧΕΙΡΙΣΗ ΚΑΙ ΑΝΑΣΥΝΘΕΣΗ ΑΡΧΕΙΩΝ ΔΙΣΚΕΤΩΝ	23
9.	ΕΠΙΛΟΓΟΣ	24
10.	CREDITS – ΕΥΧΑΡΙΣΤΙΕΣ	25

ΕΙΣΑΓΩΓΗ

Ο Amstrad CPC (Colour Personal Computer) είναι ένας από τους πιο γνωστούς 8-bit home computers τής δεκαετίας του '80. Διαθέτει τον επεξεργαστή Zilog Z80A και μνήμη από 64 Kb έως 128 Kb, με δυνατότητα επέκτασης στα 512 KB. Υποστηρίζει τρεις αναλύσεις οθόνης: 160x200 px με 16 χρώματα και 20 στήλες χαρακτήρων, 320x200 px με 4 χρώματα και 40 στήλες χαρακτήρων και 640x200 px με 2 χρώματα και 80 στήλες χαρακτήρων, ενώ ο μέγιστος αριθμός χρωμάτων τής παλέτας είναι 4096.

Σε αντίθεση με τις κονσόλες τής εποχής, ο Amstrad δεν “τρέχει” ROMs, αλλά προγράμματα που ανάλογα με το μοντέλο φορτώνονται από κασέτα, δισκέτα ή cartridge και εκτελούνται μέσα στο περιβάλλον του υπολογιστή.

Στο ROM του περιλαμβάνει την Locomotive BASIC, η οποία ενεργοποιείται αυτόματα με την εκκίνηση του συστήματος.

Τα παιχνίδια για Amstrad μπορεί να είναι:

- εξ ολοκλήρου γραμμένα σε BASIC,
- ή να χρησιμοποιούν BASIC μόνο για το αρχικό μέρος (φόρτωση, τίτλος, initialization),
- ή να είναι καθαρά σε assembly, δηλαδή σε γλώσσα μηχανής Z80.

Αυτή η ιδιαιτερότητα είναι σημαντική για τον μεταφραστή, γιατί επηρεάζει τον τρόπο που αποθηκεύονται τα δεδομένα και πώς μπορούμε να τα αλλάξουμε.

ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ ΚΑΙ ΠΕΡΙΕΧΟΜΕΝΑ

Τα παιχνίδια τού Amstrad κυκλοφορούν συνήθως ως αρχεία κασέτας (.TAP, .CDT), αρχεία δισκέτας (.DSK), ή αρχεία cartridge (.CPR).

Αυτά τα αρχεία λειτουργούν ως containers, που περιέχουν διάφορα επιμέρους αρχεία:

- προγράμματα BASIC (.BAS),
- εκτελέσιμα αρχεία μηχανικού κώδικα (.BIN),
- εικόνες (.SCR),
- ή άλλα αρχεία.

Ο Amstrad CPC διαθέτει σύστημα αρχείων που επιτρέπει διαγραφή, επανεγγραφή και ακόμη και undelete. Έτσι, μέσα σε ένα .DSK μπορεί να υπάρχουν υπολείμματα παλαιών δεδομένων ή κομμάτια από strings που φαίνονται να επαναλαμβάνονται, χωρίς να ανήκουν στο ενεργό πρόγραμμα.

Αυτό σημαίνει πως παρόλο που η απευθείας επεξεργασία και αλλαγή των δεδομένων στο αρχείο .DSK είναι εφικτή σε κάποιον βαθμό, αποτελεί παγίδα καθώς τα δεδομένα μπορεί να αλλάζουν διεύθυνση στο αρχείο, ή να επαναλαμβάνονται.

Γι' αυτόν τον λόγο, η σωστή διαδικασία είναι:

1. Να εξάγουμε τα επιμέρους αρχεία.
2. Να κάνουμε την επεξεργασία (π.χ. κείμενο, γραφικά, assembly).
3. Να τα επανατοποθετούμε στο αρχείο container (π.χ. .DSK).

Η εξαγωγή και επανατοποθέτηση των αρχείων περιγράφεται σε επόμενη ενότητα του tutorial.

ΓΡΑΜΜΑΤΟΣΕΙΡΑ ΚΑΙ ΚΩΔΙΚΟΠΟΙΗΣΗ

Ο Amstrad διαθέτει στο ROM του ένα προκαθορισμένο font, δηλαδή το σύνολο των λατινικών γραμμάτων, αριθμών και συμβόλων που εμφανίζονται στην οθόνη, που βασίζεται στην κωδικοποίηση ASCII-1967.



Η πρωτότυπη γραμματοσειρά περιλαμβάνει ορισμένα ελληνικά γράμματα, ωστόσο αυτά είναι ελλιπή και κακοσχεδιασμένα κι ως εκ τούτου πρέπει να σχεδιαστούν εξ αρχής, μαζί με όσα άλλα δεν υπάρχουν. Όπως συμβαίνει με τις τεχνικές ROM hacking, η αλλαγμένη γραμματοσειρά δεν χρειάζεται να έχει συγκεκριμένη κωδικοποίηση. Ωστόσο, για συγκεκριμένους λόγους, στο παρόν tutorial προτείνεται μία κοινή κωδικοποίηση για τους ελληνικούς χαρακτήρες, ένα ανεπίσημο πρότυπο που θα μπορεί να χρησιμοποιηθεί με τον ίδιο τρόπο σε πολλές μεταφράσεις παιχνιδιών (ενώ θα μπορεί να διαφοροποιείται, όταν χρειάζεται).



Η σχεδίαση των γραμμάτων δεν γίνεται με tile editor αλλά με εντολές, πράγμα που καθιστά τη μετάφραση πιο προγραμματιστική διαδικασία. Η BASIC παρέχει την εντολή SYMBOL, η οποία επιτρέπει στον προγραμματιστή να “ζωγραφίσει” νέους χαρακτήρες ή να αντικαταστήσει υπάρχοντες. Κάθε χαρακτήρας αποτελείται από 8 bytes (8 γραμμές x 8 pixels), και αποθηκεύεται σε ειδική περιοχή της RAM.

Η εντολή SYMBOL δέχεται 9 ορίσματα, την τιμή ASCII τού χαρακτήρα και τα 8 bytes.

Τα γράμματα στο ελληνικό και στο λατινικό αλφάβητο που έχουν κοινή σχεδίαση, είναι προτιμότερο να παραμείνουν στις θέσεις που βρίσκονται.

Αυτά είναι τα Α Β Ε Ζ Η Ι Κ Μ Ν Ο Ρ Τ Υ Χ η ο ν.

Στο πρωτότυπο αλφάβητο συμπεριλαμβάνονται και box-drawing χαρακτήρες που μπορούν να χρησιμοποιηθούν για εμφάνιση γραφικών (semigraphics):



Καθώς η πιθανότητα αυτοί οι χαρακτήρες να χρησιμοποιούνται σε κάποια παιχνίδια (για σχεδίαση γραφικών) είναι κάπως μεγαλύτερη από άλλους, είναι προτιμότερο να διατηρηθούν στη γραμματοσειρά, οπότε τα ελληνικά γράμματα τοποθετήθηκαν σε άλλες θέσεις.

Καθώς στην πρωτότυπη γραμματοσειρά ήδη υπάρχουν 16 ελληνικά γράμματα, αυτά τα συγκεκριμένα θα επανασχεδιαστούν καλύτερα στις ίδιες θέσεις.

Για τη σωστή σχεδίαση των ελληνικών γραμμάτων στο στυλ των πρωτότυπων, παρέχονται έτοιμες οι εντολές SYMBOL. Στην παρακάτω λίστα βρίσκονται οι εντολές για όλα τα γράμματα, ωστόσο σε μια μετάφραση παιχνιδιού το πιθανότερο είναι να μην απαιτούνται όλα. Η βέλτιστη πρακτική είναι να μην συμπεριληφθούν τα γράμματα που δεν χρησιμοποιούνται, ώστε να χρησιμοποιείται λιγότερη μνήμη για τους νέους χαρακτήρες.

Στην πρώτη στήλη της λίστας φαίνεται το ελληνικό γράμμα, στην δεύτερη η τιμή του σε δεκαεξαδική μορφή στην κωδικοποίηση που επιλέχθηκε και μετά ακολουθεί η εντολή SYMBOL:

Ÿ A8 SYMBOL 168,90,66,102,60,24,24,60,0
 Ĩ A9 SYMBOL 169,90,24,24,24,24,24,0
 ů AA SYMBOL 170,42,0,102,102,102,102,60,0
 ĩ AB SYMBOL 171,42,0,24,24,24,24,12,0
 Ω AC SYMBOL 172,190,227,99,99,54,20,119,0
 α B0 SYMBOL 176,0,0,122,204,204,204,122,0
 β B1 SYMBOL 177,28,50,102,124,102,102,60,0
 γ B2 SYMBOL 178,0,0,102,102,102,60,24,24
 δ B3 SYMBOL 179,126,96,60,102,102,102,60,0
 ε B4 SYMBOL 180,0,0,60,98,48,98,60,0
 θ B5 SYMBOL 181,60,102,102,126,102,102,60,0
 λ B6 SYMBOL 182,124,6,62,102,102,102,102,0
 μ B7 SYMBOL 183,0,0,102,102,102,102,125,96
 π B8 SYMBOL 184,0,0,126,102,102,102,102,0
 σ B9 SYMBOL 185,0,0,126,204,204,204,120,0
 φ BA SYMBOL 186,0,0,76,214,214,214,124,16
 ψ BB SYMBOL 187,0,0,214,214,214,124,16,16
 χ BC SYMBOL 188,0,0,198,108,56,56,108,198
 ω BD SYMBOL 189,0,0,108,198,214,214,108,0
 Σ BE SYMBOL 190,126,96,48,24,48,96,126,0
 Ω BF SYMBOL 191,124,198,198,198,108,40,238,0
 Γ E0 SYMBOL 224,126,98,96,96,96,96,0
 Δ E1 SYMBOL 225,24,60,102,102,102,102,126,0
 Θ E2 SYMBOL 226,56,108,198,254,198,198,56,0
 Λ E3 SYMBOL 227,24,60,102,102,102,102,102,0
 Ξ E4 SYMBOL 228,126,0,0,60,0,0,126,0
 Π E5 SYMBOL 229,126,102,102,102,102,102,102,0
 Φ E6 SYMBOL 230,16,124,214,214,124,16,16,0
 Ψ E7 SYMBOL 231,214,214,214,124,16,16,56,0
 Α E8 SYMBOL 232,152,188,102,102,126,102,102,0
 Ξ E9 SYMBOL 233,191,177,52,60,52,49,63,0
 Η EA SYMBOL 234,179,179,51,63,51,51,51,0
 Ξ EB SYMBOL 235,188,152,24,24,24,24,60,0
 Ο EC SYMBOL 236,156,182,99,99,99,54,28,0
 Υ ED SYMBOL 237,179,179,51,30,12,12,30,0
 ζ EE SYMBOL 238,60,12,24,24,24,12,4,8
 η EF SYMBOL 239,0,0,188,102,102,102,102,0
 ι F0 SYMBOL 240,0,0,24,24,24,24,12,0

κ F1 SYMBOL 241,0,0,102,108,120,108,102,0
ξ F2 SYMBOL 242,60,12,24,12,24,12,4,8
ρ F3 SYMBOL 243,0,0,60,102,102,102,124,96
τ F4 SYMBOL 244,0,0,126,24,24,24,12,0
υ F5 SYMBOL 245,0,0,102,102,102,102,60,0
ς F6 SYMBOL 246,0,0,60,98,96,60,6,12
ά F7 SYMBOL 247,16,0,122,204,204,204,122,0
έ F8 SYMBOL 248,8,0,60,98,48,98,60,0
ή F9 SYMBOL 249,16,0,188,102,102,102,102,0
ί FA SYMBOL 250,8,0,24,24,24,24,12,0
ό FB SYMBOL 251,8,0,60,102,102,102,60,0
ύ FC SYMBOL 252,8,0,102,102,102,102,60,0
ώ FD SYMBOL 253,16,0,108,198,214,214,108,0
ϊ FE SYMBOL 254,36,0,24,24,24,24,12,0
ϋ FF SYMBOL 255,36,0,102,102,102,102,60,0

Για ευκολία χρήσης, ακολουθεί η λίστα μόνο με τις εντολές SYMBOL:

SYMBOL 168,90,66,102,60,24,24,60,0
SYMBOL 169,90,24,24,24,24,24,0
SYMBOL 170,42,0,102,102,102,102,60,0
SYMBOL 171,42,0,24,24,24,24,12,0
SYMBOL 172,190,227,99,99,54,20,119,0
SYMBOL 176,0,0,122,204,204,204,122,0
SYMBOL 177,28,50,102,124,102,102,60,0
SYMBOL 178,0,0,102,102,102,60,24,24
SYMBOL 179,126,96,60,102,102,102,60,0
SYMBOL 180,0,0,60,98,48,98,60,0
SYMBOL 181,60,102,102,126,102,102,60,0
SYMBOL 182,124,6,62,102,102,102,102,0
SYMBOL 183,0,0,102,102,102,102,125,96
SYMBOL 184,0,0,126,102,102,102,102,0
SYMBOL 185,0,0,126,204,204,204,120,0
SYMBOL 186,0,0,76,214,214,214,124,16
SYMBOL 187,0,0,214,214,214,124,16,16
SYMBOL 188,0,0,198,108,56,56,108,198
SYMBOL 189,0,0,108,198,214,214,108,0
SYMBOL 190,126,96,48,24,48,96,126,0
SYMBOL 191,124,198,198,198,108,40,238,0
SYMBOL 224,126,98,96,96,96,96,96,0
SYMBOL 225,24,60,102,102,102,102,126,0
SYMBOL 226,56,108,198,254,198,198,56,0
SYMBOL 227,24,60,102,102,102,102,102,0
SYMBOL 228,126,0,0,60,0,0,126,0
SYMBOL 229,126,102,102,102,102,102,102,0
SYMBOL 230,16,124,214,214,124,16,16,0
SYMBOL 231,214,214,214,124,16,16,56,0
SYMBOL 232,152,188,102,102,126,102,102,0
SYMBOL 233,191,177,52,60,52,49,63,0
SYMBOL 234,179,179,51,63,51,51,51,0
SYMBOL 235,188,152,24,24,24,24,60,0
SYMBOL 236,156,182,99,99,99,54,28,0
SYMBOL 237,179,179,51,30,12,12,30,0
SYMBOL 238,60,12,24,24,24,12,4,8
SYMBOL 239,0,0,188,102,102,102,102,0

SYMBOL 240,0,0,24,24,24,24,12,0
SYMBOL 241,0,0,102,108,120,108,102,0
SYMBOL 242,60,12,24,12,24,12,4,8
SYMBOL 243,0,0,60,102,102,102,124,96
SYMBOL 244,0,0,126,24,24,24,12,0
SYMBOL 245,0,0,102,102,102,102,60,0
SYMBOL 246,0,0,60,98,96,60,6,12
SYMBOL 247,16,0,122,204,204,204,122,0
SYMBOL 248,8,0,60,98,48,98,60,0
SYMBOL 249,16,0,188,102,102,102,102,0
SYMBOL 250,8,0,24,24,24,24,12,0
SYMBOL 251,8,0,60,102,102,102,60,0
SYMBOL 252,8,0,102,102,102,102,60,0
SYMBOL 253,16,0,108,198,214,214,108,0
SYMBOL 254,36,0,24,24,24,24,12,0
SYMBOL 255,36,0,102,102,102,102,60,0

Η χρήση της εντολής SYMBOL απαιτεί πρώτα την εντολή SYMBOL AFTER.

Πρόκειται για εντολή που ουσιαστικά καθορίζει τη μνήμη από την οποία και μετά θα χρησιμοποιούνται οι custom χαρακτήρες που δημιουργούνται από τις επόμενες εντολές SYMBOL. Στην πραγματικότητα, οι νέοι χαρακτήρες δεν αντικαθιστούν τους πρωτότυπους στη μνήμη, αλλά γράφονται στη μνήμη επιπλέον, σε νέα διεύθυνση. Η εντολή SYMBOL AFTER καθοδηγεί το πρόγραμμα στο να παίρνει τον χαρακτήρα που πρόκειται να τυπώσει από την περιοχή μνήμης των πρωτότυπων χαρακτήρων, ή αυτή των επανασχεδιασμένων. Μάλιστα εάν επαναλάβουμε εντολή SYMBOL για κάποιον custom χαρακτήρα που έχουμε σχεδιάσει, η δεύτερη σχεδίαση γράφεται σε τρίτη θέση στη μνήμη. Αυτό σημαίνει ότι θα ήταν καλό να περιορίσουμε τη χρήση των custom χαρακτήρων, όσο είναι δυνατόν. Π.χ. τα γράμματα που βρίσκονται πρώτα στη λίστα (ÿ, ï, û, ð, Ω) και τα οποία είναι πολύ πιθανό να μην χρειαστούν στη μετάφραση του παιχνιδιού, καλό θα ήταν να μην συμπεριληφθούν, αν όντως δεν θα χρησιμοποιηθούν στο κείμενο. Εάν το πρώτο γράμμα που θα συμπεριληφθεί είναι το α (που έχει τιμή B0 / 176), θα πρέπει να προηγηθεί η εντολή SYMBOL AFTER 176. Εάν χρειάζεται να συμπεριληφθεί το γράμμα ÿ (που έχει τιμή A8 / 168), θα πρέπει να προηγηθεί η εντολή SYMBOL AFTER 168.

Η εντολή SYMBOL AFTER ωστόσο σημαίνει πως όλοι οι επόμενοι χαρακτήρες της γραμματοσειράς θα είναι custom, είτε τους επανασχεδιάζουμε είτε όχι. Δηλαδή ακόμα και ο χαρακτήρας ■ π.χ., που δεν επανασχεδιάζουμε στην ελληνική γραμματοσειρά, αντιγράφεται εκ νέου στη θέση μνήμης των custom χαρακτήρων και θα καταναλώνει 8 bytes – απλώς θα έχει την ίδια σχεδίαση με τον πρωτότυπο.

Εάν έχουμε χρησιμοποιήσει π.χ. SYMBOL AFTER 168 και έχουμε εμφανίσει ελληνικό κείμενο, δίνοντας SYMBOL AFTER 0 ουσιαστικά πια θα χρησιμοποιούμε την πρωτότυπη γραμματοσειρά και τα ελληνικά γράμματα που είχαμε σχεδιάσει με τις εντολές SYMBOL απλώς δεν θα χρησιμοποιούνται.

Εάν το παιχνίδι ξεκινάει τρέχοντας ένα αρχείο .BAS, μπορούμε με τις παραπάνω εντολές SYMBOL να τυπώσουμε ελληνικό κείμενο. Ωστόσο μπορεί το παιχνίδι στη συνέχεια να φορτώνει στη μνήμη ένα αρχείο .BIN, με γλώσσα μηχανής Z80. Όταν θα φορτώνεται αυτό, είναι πολύ πιθανό να γράφονται άλλα δεδομένα στη θέση των bytes των custom χαρακτήρων. Ο assembly κώδικας του παιχνιδιού που βρίσκεται στο .BIN παίρνει υπόψη του την τελευταία εντολή SYMBOL AFTER και εάν δεν έχει αντιγράψει άλλα bytes στις θέσεις των custom χαρακτήρων, θα τους τυπώνει κανονικά. Εάν όμως έχει αντιγράψει άλλα bytes στις θέσεις των custom χαρακτήρων, θα τυπώνει τους χαρακτήρες από τις θέσεις μνήμης των custom χαρακτήρων ενώ εκεί θα έχουν γραφτεί άλλα δεδομένα. Το αποτέλεσμα θα είναι να εμφανίζονται στην οθόνη pixels που αντιστοιχούν σε άλλα bytes, οπότε θα φαίνονται σαν σκουπίδια.

Στην περίπτωση που θέλουμε να τυπώσουμε τα ελληνικά γράμματα σωστά αλλά συμβαίνει το παραπάνω, θα πρέπει ακριβώς πριν την εκτύπωση να αντιγράψουμε σε αυτές τις θέσεις μνήμης τα bytes των ελληνικών χαρακτήρων και αμέσως μετά να επαναφέρουμε τα bytes τού προγράμματος. Η διαδικασία που θα πρέπει να ακολουθήσουμε, έχει τα εξής βήματα:

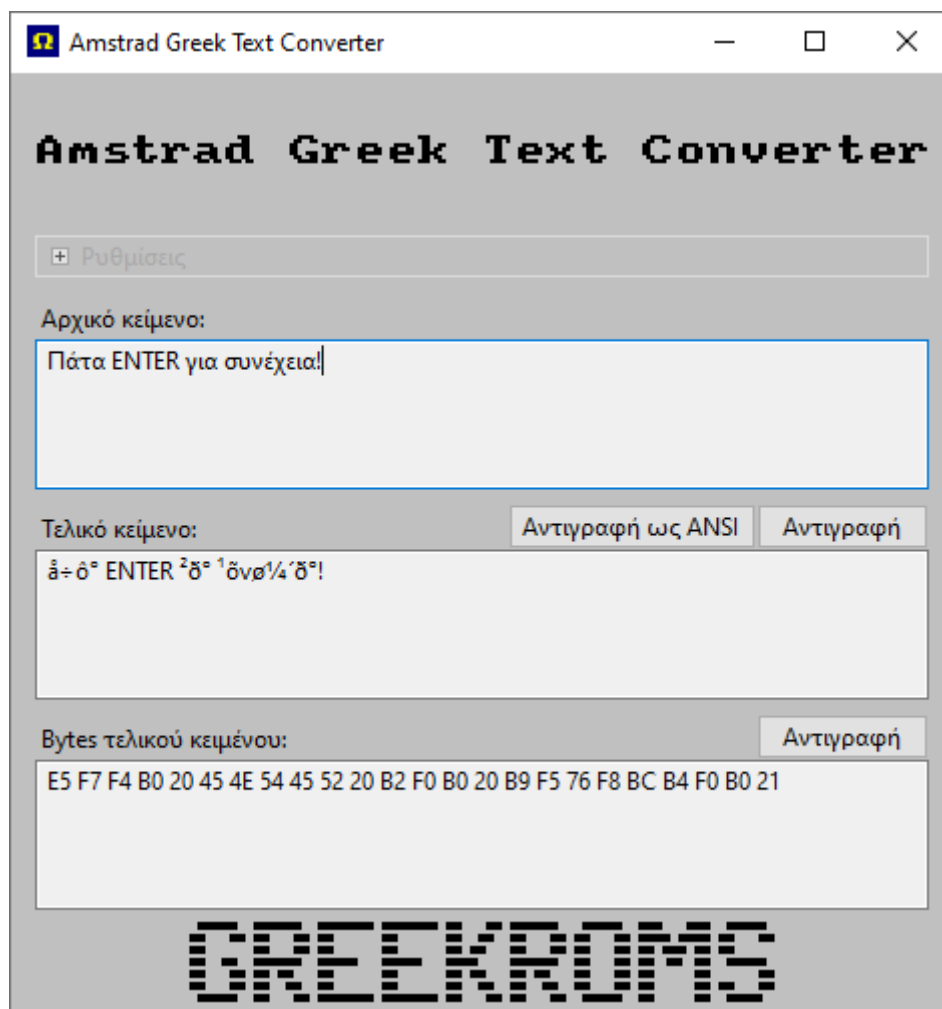
1. Πρέπει να εντοπίσουμε τις θέσεις μνήμης όπου έχουν γραφτεί οι custom χαρακτήρες. Αυτό μπορούμε να το επιτύχουμε απλώς αναζητώντας στη μνήμη (όπως αυτή είναι σε κάποιο σημείο τού παιχνιδιού που ακόμα δεν έχει φορτωθεί το αρχείο BIN, αλλά τρέχουν εντολές BASIC) τις τιμές των χαρακτήρων μας. Π.χ. αν θέλουμε να βρούμε πού βρίσκεται το α, θα πρέπει να αναζητήσουμε τα bytes 0,0,122,204,204,204,122,0 (δηλαδή τις τιμές 00 00 7A CC CC CC 7A 00). Καθώς στην κωδικοποίησή μας δεν τοποθετήσαμε τα custom γράμματα με τη σειρά, τα bytes τους δεν θα βρίσκονται και στη μνήμη όλα συνεχόμενα. Αυτό θα ισχύει μόνο για όσα γράμματα είναι συνεχόμενα και στην κωδικοποίηση.
2. Πρέπει να βρούμε αρκετό διαθέσιμο χώρο στο αρχείο .BIN και στη μνήμη και να τοποθετήσουμε εκεί τα bytes των custom χαρακτήρων, αλλά και τα bytes που τοποθετούνται σε αυτές τις θέσεις, όταν φορτώνεται το αρχείο .BIN (που μπορεί να είναι κώδικας ή άλλα δεδομένα).
3. Πρέπει να προσθέσουμε δύο ρουτίνες στο πρόγραμμά μας, που να αντιγράφουν η μία τα bytes των custom χαρακτήρων στις θέσεις που βρήκαμε στο προηγούμενο βήμα και η άλλη τα bytes τού .BIN που γράφονται στις ίδιες θέσεις.
4. Πρέπει πριν κάθε εκτύπωση string, να τρέχουμε τη ρουτίνα που αντιγράφει τα bytes των custom χαρακτήρων και μετά την εκτύπωση, τη ρουτίνα που αντιγράφει τα bytes τού .BIN. Εναλλακτικά, μπορούμε να φτιάξουμε μια άλλη ρουτίνα, σαν container, που να κάνει αυτό και ενδιάμεσα αυτή να καλεί και την ρουτίνα εκτύπωσης που χρησιμοποιεί το πρόγραμμα. Την πρωτότυπη ρουτίνα εκτύπωσης, όπου χρησιμοποιούνταν, θα πρέπει να την αντικαταστήσουμε με τη δική μας.

Η παραπάνω επιλεγμένη κωδικοποίηση γραμμάτων δεν είναι κατάλληλη για παιχνίδια όπου ο παίκτης πρέπει να πληκτρολογήσει κείμενο (στα Ελληνικά). Σ' αυτή την περίπτωση, θα πρέπει οι εντολές SYMBOL να αντικαταστήσουν τα υπάρχοντα λατινικά γράμματα που βρίσκονται στις ίδιες θέσεις πλήκτρων (π.χ. το Γ πρέπει να σχεδιαστεί στη θέση τού G). Τα γράμματα με τόνους θα μπορούν να σχεδιαστούν σε οποιαδήποτε θέση, ωστόσο δεν θα μπορούν να πληκτρολογηθούν, εκτός κι αν γίνουν εκτεταμένες αλλαγές σε ρουτίνες assembly. Η λογική θα είναι η ίδια, ωστόσο το πρόγραμμα Amstrad Greek Text Converter δεν θα μπορεί να χρησιμοποιηθεί – θα πρέπει να δημιουργήσετε ένα αρχείο table με τη δική σας κωδικοποίηση και να επεξεργαστείτε το κείμενο με έναν hex editor όπως ο Thingy32.

ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΝΟΥ

Στον hex editor και καθώς η νέα κωδικοποίηση έχει τα ελληνικά γράμματα σε σειρά που διαφέρει από την ASCII κωδικοποίηση, τα strings στα Ελληνικά δεν θα είναι αναγνωρίσιμα. Θα πρέπει να εισάγουμε τις τιμές χαρακτήρων που στην κωδικοποίησή μας θα αντιστοιχούν στα γράμματα που θέλουμε, αλλά καθώς επεξεργαζόμαστε το αρχείο στα Windows ή άλλο σύγχρονο λειτουργικό σύστημα και hex editor, θα βλέπουμε τους χαρακτήρες που αντιστοιχούν σε αυτές τις ASCII τιμές.

Για να είναι ευκολότερη, γρηγορότερη και ασφαλέστερη η μετατροπή τού κειμένου, προτείνεται η χρήση τού προγράμματος Amstrad Greek Text Converter.



Απλώς πληκτρολογήστε τό κείμενο που θέλετε στα Ελληνικά και αντιγράψτε στο αρχείο τα bytes για το αντίστοιχο string, ώστε στο παιχνίδι να εμφανίζεται σωστά το κείμενο. Δείχνει επίσης και το ASCII κείμενο γι' αυτά τα bytes, ωστόσο αυτό με τους περισσότερους hex editors δεν μπορείτε να το περάσετε στο αρχείο τού παιχνιδιού.

Εναλλακτικά, μπορείτε να επεξεργαστείτε το αρχείο με έναν hex editor που υποστηρίζει tables, όπως ο Thingy32, χρησιμοποιώντας τό παρακάτω table (μπορείτε και να το κατεβάσετε από εδώ: <https://greekroms.net/files/Amstrad.tbl>):

20=
21=!
22=" "
23=#
24=\$
25=%
26=&
27=' '
28=(
29=)
2A=*
2B=+
2C=,
2D=-
2E=.
2F=/
30=0
31=1
32=2
33=3
34=4
35=5
36=6
37=7
38=8
39=9
3A=:
3B=;
3C=<
3D==
3E=>
3F=?
40=@
41=A
42=B

43=C
44=D
45=E
46=F
47=G
48=H
49=I
4A=J
4B=K
4C=L
4D=M
4E=N
4F=O
50=P
51=Q
52=R
53=S
54=T
55=U
56=V
57=W
58=X
59=Y
5A=Z
5B=[
5C=\
5D=]
5F=_
60=`
61=a
62=b
63=c
64=d
65=e
66=f
67=g
68=h
69=i

$6A=j$
 $6B=k$
 $6C=l$
 $6D=m$
 $6E=n$
 $6F=o$
 $70=p$
 $71=q$
 $72=r$
 $73=s$
 $74=t$
 $75=u$
 $76=v$
 $77=w$
 $78=x$
 $79=y$
 $7A=z$
 $7B=\{$
 $7C=|$
 $7D=\}$
 $7E=\sim$
 $90=\cdot$
 $A8=\ddot{Y}$
 $A9=\ddot{I}$
 $AA=\ddot{u}$
 $AB=\ddot{t}$
 $AC=\Omega$
 $B0=\alpha$
 $B1=\beta$
 $B2=\gamma$
 $B3=\delta$
 $B4=\varepsilon$
 $B5=\theta$
 $B6=\lambda$
 $B7=\mu$
 $B8=\pi$
 $B9=\varsigma$
 $BA=\varphi$

BB= ψ
 BC= χ
 BD= ω
 BE= Σ
 BF= Ω
 E0= Γ
 E1= Δ
 E2= Θ
 E3= Λ
 E4= Ξ
 E5= Π
 E6= Φ
 E7= Ψ
 E8= 'A
 E9= 'E
 EA= 'H
 EB= 'I
 EC= 'O
 ED= 'Y
 EE= ζ
 EF= η
 F0= ι
 F1= κ
 F2= ξ
 F3= ρ
 F4= τ
 F5= υ
 F6= ς
 F7= $\acute{\alpha}$
 F8= $\acute{e}$
 F9= $\acute{\eta}$
 FA= $\acute{\iota}$
 FB= $\acute{o}$
 FC= $\acute{u}$
 FD= $\acute{\omega}$
 FE= $\ddot{\imath}$
 FF= $\ddot{u}$

ΑΛΛΑΓΗ ΓΡΑΦΙΚΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

Τεχνικές πληροφορίες

Ο Amstrad CPC υποστηρίζει τρεις αναλύσεις οθόνης και μία ανεπίσημη:

- Mode 0: 160x200 px με 16 χρώματα (4 bpp)
- Mode 1: 320x200 px με 4 χρώματα (2 bpp)
- Mode 2: 640x200 px με 2 χρώματα (1 bpp)
- Mode 3: 160x200 px με 2 χρώματα (2bpp) (μη επίσημη λειτουργία, αποτέλεσμα συγκεκριμένου hardware glitch)

Με τη χρήση τεχνικής overscan, η ορατή περιοχή της οθόνης μπορεί να επεκταθεί πέρα από τις τυπικές διαστάσεις. Το overscan απαιτεί συνήθως custom ρουτίνες, όχι BASIC εντολές και συναντάται πιο συχνά σε demos.

Οι πιο συνηθισμένες overscan αναλύσεις είναι:

- Full screen mode 0: 192x272 px με 16 χρώματα (4 bpp)
- Full screen mode 1: 384x272 px με 4 χρώματα (2 bpp)
- Full screen mode 2: 768x272 px με 2 χρώματα (1 bpp)

Η περιοχή μνήμης της οθόνης (video RAM) ξεκινά από τη διεύθυνση C000 (ή 49152 σε δεκαδικό) και εκτείνεται έως το FFFF. Εκεί βρίσκονται τα bytes που καθορίζουν τι εμφανίζεται στην οθόνη.

Αρχεία .SCR

Οι στατικές εικόνες που εμφανίζονται στο παιχνίδι, πιθανόν να είναι ενσωματωμένες στο αρχείο .BIN του παιχνιδιού, ή να αποθηκεύονται σε ανεξάρτητα αρχεία .SCR.

Τα αρχεία .SCR είναι στην ουσία αποτυπώσεις της μνήμης οθόνης. Περιλαμβάνουν είτε τα raw bytes που γράφονται στη μνήμη οθόνης, είτε – σε ορισμένες περιπτώσεις – και πρόσθετες πληροφορίες, όπως header (π.χ. μήκος αρχείου, διεύθυνση φόρτωσης), παλέτα χρωμάτων, ή ακόμη και metadata για editors.

Συνήθως, ένα “καθαρό” .SCR περιέχει ακριβώς 4000 / 16384 bytes (για τα modes 0/1/2) – δηλαδή τα δεδομένα που φορτώνονται απευθείας στη μνήμη της οθόνης. Αν όμως το αρχείο είναι μεγαλύτερο, υπάρχει πιθανότητα να έχει header ή παλέτα.

Για να το χρησιμοποιήσουμε σε παιχνίδι, μπορεί να θέλουμε μόνο τα raw data (χωρίς headers ή επιπλέον bytes). Αυτό σημαίνει ότι ίσως χρειαστεί να τα αφαιρέσουμε ή να τα εξαγάγουμε με τον hex editor.

Για να αλλάξετε μία εικόνα .SCR, όπως π.χ. μία loading screen, είτε είναι σε ανεξάρτητο αρχείο .SCR, είτε τα bytes βρίσκονται σε κάποια διεύθυνση του αρχείου .BIN, θα πρέπει να δημιουργήσετε ή να επεξεργαστείτε την εικόνα που θέλετε και να την μετατρέψετε σε αρχείο .SCR.

Επεξεργασία εικόνας

Γνωρίζοντας τα modes γραφικών τού Amstrad, μπορείτε να επεξεργαστείτε μία εικόνα με κάποιο σύγχρονο πρόγραμμα επεξεργασίας εικόνων όπως το Gimp, καθορίζοντας την παλέτα του ώστε να έχει τον κατάλληλο αριθμό χρωμάτων και τα συγκεκριμένα χρώματα που ανήκουν και στην παλέτα τού Amstrad. Κατά την επεξεργασία της εικόνας, η καταλληλότερη μορφή είναι αρχείο .PNG, καθώς υποστηρίζει παλέτα με τον αριθμό χρωμάτων που θέλετε, χρησιμοποιεί lossless συμπίεση (δεν χαλάει η ποιότητα της εικόνας κατά την αποθήκευση, όπως με τα αρχεία .JPG) και υποστηρίζεται από όλα τα σύγχρονα προγράμματα.

Η τελική μετατροπή σε μορφή .SCR μπορεί να γίνει εύκολα με το πρόγραμμα Multipaint (multipaint.kameli.net), ή μπορείτε εξ αρχής να επεξεργαστείτε την εικόνα με αυτό. Το αρχείο .SCR που παράγει το Multipaint διαθέτει header και πληροφορίες παλέτας, δηλαδή bytes πριν και μετά τα raw data που ενδεχομένως να χρειάζεστε μόνο τους, οπότε θα πρέπει να κρατήσετε μόνο τα 16.384 bytes που αντιστοιχούν στα raw data, αντιγράφοντάς τα με τον hex editor. Αυτά αναγνωρίζονται σχετικά εύκολα, γνωρίζοντας και το σωστό μήκος, που είναι όσο και η μνήμη video RAM (&4000 / 16384 bytes). Ωστόσο σε αρχεία διαφορετικών modes βρίσκονται σε διαφορετικά σημεία.

Εισαγωγή εικόνας στο αρχείο

Στην περίπτωση που τα γραφικά είναι ενσωματωμένα στο αρχείο .BIN και δεν βρίσκονται σε ξεχωριστό αρχείο .SCR, θα πρέπει να τα εντοπίσετε. Ωστόσο υπάρχει η πιθανότητα να είναι συμπιεσμένα, ή να σχεδιάζονται με διαφορετικό τρόπο, π.χ. με tileset και tilemap, για καλύτερη οικονομία χώρου στο αρχείο (όπως συμβαίνει συχνά στα commercial games). Και στις δύο αυτές περιπτώσεις, υπάρχουν δύο λύσεις. Η πρώτη λύση είναι να ακολουθήσετε την αρχική λογική. Δηλαδή θα βρείτε την πρωτότυπη συμπίεση και (εάν την αναγνωρίσετε και βρείτε αντίστοιχο πρόγραμμα

συμπίεσης) συμπίεσετε τα δεδομένα γραφικών, ή, στην περίπτωση tileset και tilemap (που μπορεί επιπλέον να χρησιμοποιεί και συμπίεση), να ξεχωρίσετε τα tiles της δικής σας εικόνας και να φτιάξετε αντίστοιχο tileset και tilemap. Η δεύτερη και πιο καθαρή λύση, η οποία καθιστά τις μελλοντικές αλλαγές στα γραφικά πιο απλές και επαναλήψιμες, είναι να βρείτε τη ρουτίνα αποσυμπίεσης ή δημιουργίας των γραφικών από το tileset και το tilemap και να την αντικαταστήσετε από μια άλλη, γνωστή ρουτίνα αποσυμπίεσης όπως ZX0 (<https://github.com/einar-saukas/ZX0>) ή ZX1 (<https://github.com/einar-saukas/ZX1>), καθώς είναι αρκετά πιο αποδοτική και τα συμπίεσμένα δεδομένα θα χωράνε ευκολότερα στο αρχείο.

Σε πολλές περιπτώσεις, η λογική και ο αλγόριθμος που θα έχει χρησιμοποιηθεί στο παιχνίδι, μπορεί να είναι μοναδικά, οπότε να τα αντιμετωπίζετε ως ξεχωριστές περιπτώσεις.

Παράδειγμα φόρτωσης εικόνας .SCR με εντολές BASIC

Αν έχετε δημιουργήσει ένα αρχείο TITLE.SCR, το οποίο περιέχει τα raw data και όχι header και λοιπά δεδομένα, μπορείτε να το εμφανίσετε στον Amstrad με τις παρακάτω εντολές BASIC:

```
MODE 1
```

```
LOAD "TITLE.SCR", &C000
```

ΔΙΑΤΑΞΗ ΜΝΗΜΗΣ, ΡΟΥΤΙΝΕΣ ΣΥΣΤΗΜΑΤΟΣ, BASIC

Κατά τη διαδικασία τής μετάφρασης, θα χρειαστεί να επεξεργαστείτε το BASIC listing, είτε αλλάζοντας απλές και προφανείς BASIC εντολές όπως π.χ. η PRINT, είτε εντολές που απαιτούν περισσότερες γνώσεις. Η Locomotive BASIC, εκτός από γλώσσα προγραμματισμού, λειτουργεί και ως περιβάλλον φόρτωσης και εκτέλεσης προγραμμάτων. Ακόμη και πολλά παιχνίδια σε assembly χρησιμοποιούν BASIC loader που εμφανίζει μηνύματα ή επιλογές, τα οποία είναι συχνά τα πρώτα που θα χρειαστεί να μεταφράσετε.

Στις εντολές assembly θα συναντήσετε CALLs σε ρουτίνες εκτύπωσης χαρακτήρων ή strings που θα σας είναι άγνωστες. Οι διευθύνσεις μνήμης τους συχνά υπερβαίνουν το μέγεθος του αρχείου .BIN, που σημαίνει ότι δεν βρίσκονται μέσα στο ίδιο το αρχείο αλλά στο ROM του Amstrad και θα μπορείτε να τις δείτε στη μνήμη όταν εκτελείται το πρόγραμμα. Οι ρουτίνες αυτές, γνωστές και ως firmware routines, περιλαμβάνουν λειτουργίες για εκτύπωση χαρακτήρων, ανάγνωση πληκτρολογίου, διαχείριση γραφικών, ήχου και άλλες βασικές λειτουργίες του συστήματος. Οι ρουτίνες αυτές του συστήματος είναι καταγεγραμμένες και μπορείτε να βρείτε πληροφορίες γι' αυτές, ώστε να μπορείτε να τις κατανοήσετε και να τις αξιοποιήσετε.

Η γνώση και η εμβάθυνση στη διάταξη μνήμης τού Amstrad, στη Locomotive BASIC, στις ρουτίνες συστήματος αλλά και γενικότερα στη γλώσσα Z80 assembly και στη γλώσσα μηχανής διευκολύνει κατά πολύ τη διαδικασία hacking και μετάφρασης ενός παιχνιδιού και δίνει τη δυνατότητα περισσότερων αλλαγών και προσθηκών. Το συγκεκριμένο tutorial δεν έχει σκοπό την κάλυψη των συγκεκριμένων τομέων, ωστόσο θα χρειαστεί να αποκτήσετε έστω γενικές γνώσεις τους. Για όσους θέλουν να εμβαθύνουν περισσότερο, οι παρακάτω πηγές περιέχουν αναλυτικές πληροφορίες για τη διάταξη μνήμης, τις ρουτίνες firmware και τη Locomotive BASIC:

Διάταξη μνήμης

https://www.cpcwiki.eu/index.php/Amstrad_Whole_Memory_Guide

https://www.cpcalive.com/docs/amstrad_cpc_6128_memory_map.html

<https://www.chibiakumas.com/z80/AmstradCPC.php>

Ρουτίνες συστήματος

<https://www.scribd.com/document/675462182/Amstrad-CPC464-Firmware-ROM-Routines-and-Explanations-ENGLISH>

https://cpcrulez.fr/codingBOOK_soft968-CPC464-664-6128_firmware_001.htm

Locomotive BASIC

https://en.wikipedia.org/wiki/Locomotive_BASIC

https://www.cpcwiki.eu/index.php/Locomotive_BASIC

<https://www.qsl.net/hb9xch/computer/amstrad/locomotivebasic.html>

https://rosettacode.org/wiki/Category:Locomotive_Basic

https://web.archive.org/web/20190604085927fw_/http://cpctech.cpc-live.com/docs/bastech.html

ΔΙΑΧΕΙΡΙΣΗ ΚΑΙ ΑΝΑΣΥΝΘΕΣΗ ΑΡΧΕΙΩΝ ΔΙΣΚΕΤΩΝ

Όπως αναφέρθηκε, τα αρχεία .DSK, .TAP, .CDT ή .CPR είναι απλώς “εικόνες” των δισκετών ή κασετών. Το .DSK είναι η πιο συνηθισμένη μορφή για εξομοίωση δισκέτας του Amstrad. Τα .TAP και .CDT χρησιμοποιούνται για εικόνες κασετών, ενώ το .CPR για cartridges της έκδοσης GX4000 και CPC plus. Δεν είναι πρακτικό να επεξεργαζόμαστε απευθείας τα δεδομένα μέσα σε αυτά – πρέπει πρώτα να εξάγουμε τα επιμέρους αρχεία, να τα επεξεργαζόμαστε και έπειτα να τα επανεισάγουμε. Το κατάλληλο πρόγραμμα για διαχείριση αρχείων .DSK είναι το ManageDsk:

https://www.genesis8bit.fr/archives/index.php?news_id=315

https://cpcrulez.fr/emulateurs_util-dsk-managedsk.htm

<https://github.com/cpcsdk/idsk> (έκδοση για Linux/Unix, με όνομα iDSK)

Χρήσιμο πρόγραμμα για διαχείριση αρχείων .DSK είναι και το CPCDiskXP. Με το CPCDiskXP μπορείτε να δημιουργήσετε νέα, κενά αρχεία .DSK, καθώς και να τα γράψετε σε κανονικές δισκέτες:

<https://www.cpcmania.com/cpcdiskxp/cpcdiskxp.htm>

<https://www.cpcwiki.eu/index.php/CPCDiskXP>

https://cpcrulez.fr/emulateurs_UTIL-DSK-CPCDiskXP.htm

Μετά την ολοκλήρωση της μετάφρασης, τα περιεχόμενα αρχεία του .DSK θα έχουν ξαναγραφτεί σε διαφορετικές θέσεις στα sectors της δισκέτας. Η λειτουργικότητα δεν επηρεάζεται, ωστόσο ενδεχομένως να παρατηρείται μια ανεπαίσθητη καθυστέρηση στο διάβασμα των αρχείων από το disk drive και τη δισκέτα. Ο ίδιος ο Amstrad δεν διέθετε κάποιο εργαλείο defrag, αν και είχαν κυκλοφορήσει τρίτα εργαλεία της εποχής που βελτίωναν την ταχύτητα ανάγνωσης των δισκετών.

Για βέλτιστη ταχύτητα φόρτωσης, δημιουργήστε ένα νέο, καθαρό .DSK και εισάγετε τα αρχεία με τη σειρά που επιθυμείτε. Συνήθως, αν το αρχείο εκκίνησης (loader) τοποθετηθεί πρώτο, το παιχνίδι φορτώνει αισθητά γρηγορότερα. Μπορείτε να πειραματιστείτε με τη σειρά εγγραφής των αρχείων και να συγκρίνετε την ταχύτητα ανάγνωσης με διαφορετικούς συνδυασμούς.

- Δημιουργήστε ένα νέο, κενό αρχείο .DSK με το CPCDiskXP.
- Με το CPCDiskXP ή το ManageDsk, εισάγετε στο νέο .DSK το αρχείο που θέλετε να γραφτεί στα πρώτα sectors της δισκέτας και αποθηκεύστε το.
- Επαναλάβετε για κάθε επόμενο αρχείο, αποθηκεύοντας το .DSK κάθε φορά που εισάγετε κάποιο.

ΕΠΙΛΟΓΟΣ

Ο Amstrad CPC υπήρξε ένας από τους πιο πετυχημένους home computers τής δεκαετίας τού 1980, με πλήθος παιχνιδιών και προγραμμάτων. Μέσα από τις μεταφράσεις, το έργο των προγραμματιστών διασώζεται, καθίσταται κατανοητό στις σύγχρονες κοινότητες και παραμένει ζωντανό ως μέρος τής ιστορίας των ηλεκτρονικών παιχνιδιών. Η τήρηση της αυθεντικότητας του αρχικού περιεχομένου και του αισθητικού ύφους τού δημιουργού είναι ουσιώδης για τη διατήρηση της ιστορικής και πολιτισμικής αξίας κάθε έργου.

Η μετάφραση παιχνιδιών για τον Amstrad CPC έχει αρκετά χαρακτηριστικά που τη διαφοροποιούν από τις κλασικές τεχνικές ROM hacking: τα .DSK/.TAP containers, την ενσωματωμένη Locomotive BASIC, τη σχεδίαση custom χαρακτήρων με την εντολή SYMBOL και τις ιδιαιτερότητες των modes γραφικών. Ωστόσο, από ένα σημείο και μετά η λογική και οι διαδικασίες συγκλίνουν.

Αυτό σημαίνει ότι η μετάφραση παιχνιδιών τού Amstrad CPC ανοίγει νέα μονοπάτια μάθησης (BASIC tricks, copy/restore ρουτίνες, μορφές .SCR, compressors), αλλά αξιοποιεί και την ήδη υπάρχουσα εμπειρία τού ROM hacking σε κλασικά συστήματα και κονσόλες. Όποιος έχει υπομονή και διάθεση να μάθει τις ιδιαιτερότητες του συστήματος, θα δει πως οι γνώριμες τεχνικές γίνονται τα εργαλεία του για σταθερά, επαγγελματικά αποτελέσματα, αλλά και αντιστρόφως, οι νέες τεχνικές για το hacking παιχνιδιών τού Amstrad CPC θα τον βοηθήσουν με άλλα συστήματα.

CREDITS - ΕΥΧΑΡΙΣΤΙΕΣ

Ο οδηγός αυτός γράφτηκε με σκοπό να βοηθήσει τους ROM hackers άλλων κονσολών και υπολογιστών να μεταφράσουν τα κλασικά παιχνίδια του Amstrad CPC στα Ελληνικά και να τα ζωντανέψουν ξανά, μετά από δεκαετίες.

Ιδιαίτερες ευχαριστίες σε όλους τους δημιουργούς των emulators, των επεξεργαστών και των εργαλείων που αναφέρονται εδώ· χωρίς τη δουλειά τους, τίποτα από αυτά δεν θα ήταν εφικτό.

Ευχαριστίες επίσης στις κοινότητες φίλων του Amstrad CPC, που συνεχίζουν να κρατούν ζωντανό το πνεύμα και την κληρονομιά του συστήματος.

Καλή επιτυχία και... καλές μεταφράσεις!

© 2025 GreekRoms (Vag)