

# ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΘΕΜΑ ΑΠΑΛΛΑΚΤΙΚΗΣ ΕΡΓΑΣΙΑΣ 2023-2024

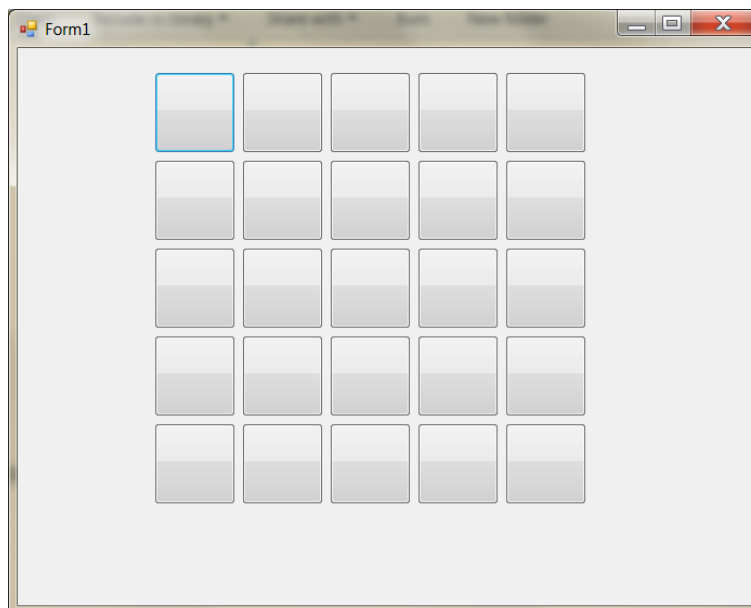
### Σύνολο μονάδων: 6

### Διδάσκων: Ευθύμιος Αλέπης

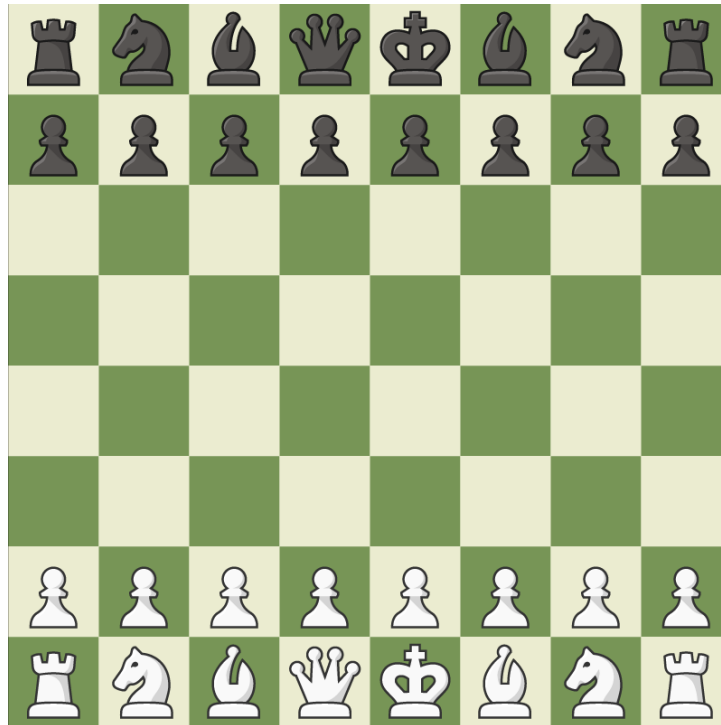
#### Περιγραφή:

Η κάθε ομάδα καλείται να αναπτύξει 3 παραθυρικές εφαρμογές (windows form applications) σε γλώσσα προγραμματισμού C#. Οι εφαρμογές αναλύονται ως ακολούθως:

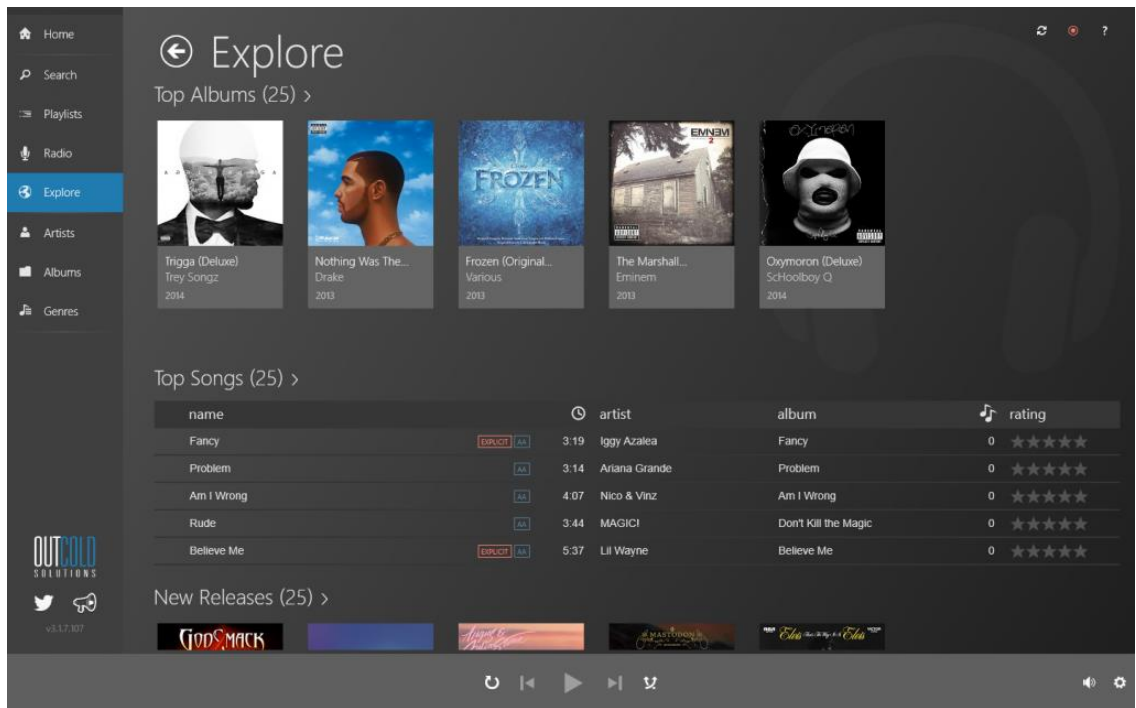
- **Εφαρμογή αντίστοιχη με την τρίλιζα αλλά σε διαστάσεις  $N \times N$ .** Βασικό επίπεδο υλοποίησης: Καλείσθε να υλοποιήσετε την γνωστή σας «τρίλιζα», ως windows form Application, σε διαστάσεις  $N > 3$ , όπου το  $N$  θα το δίνει ο χρήστης και θα πρέπει να είναι περιττός αριθμός (π.χ. 5, 7, κ.λπ.). Δεν θα σχεδιάζετε κάποιο γραφικό πάνω στη φόρμα, αλλά θα χρησιμοποιήσετε τα συνηθισμένα κουμπιά που διαθέτουν οι φόρμες (δεν είναι απαραίτητο να υπάρχουν γραμμές, ωστόσο είναι απαραίτητο τα σημεία που μπορεί να «πατήσει» κανείς για να κάνει την επιλογή του να είναι τετράγωνα κουμπιά). Δεν θα υλοποιηθεί σε διαστάσεις  $3 \times 3$ , αλλά σε διαστάσεις  $N \times N$ . Αντίστοιχα, για να κερδίσει κανείς στο παιχνίδι θα πρέπει να συγκεντρώσει  $N$  ίδιους χαρακτήρες ( $X$  ή  $0$ ), οριζοντίως, καθέτως, ή διαγωνίως. Το παιχνίδι θα παρέχει τη δυνατότητα απλά να παίζουν 2 χρήστες, όπου οι χαρακτήρες  $X$  και  $0$  θα εναλλάσσονται αναλόγως με τη σειρά πατήματος των κουμπιών. Ωστόσο, το παιχνίδι θα πρέπει να είναι σε θέση να «καταλαβαίνει» πότε κάποιος από τους 2 παίκτες έχει κερδίσει και να βγάζει κάποιο μήνυμα. Προχωρημένο επίπεδο υλοποίησης: Θα υπάρχει και η δυνατότητα ο «αντίπαλος» να είναι ο ίδιος ο υπολογιστής (το ποιες κινήσεις θα κάνει ο υπολογιστής αφήνεται στη δική σας κρίση. Ωστόσο, δεν είναι απαραίτητο, ο υπολογιστής να παίζει «έξυπνα», μπορεί απλά να παίζει «τυχαία»). Πάλι θα πρέπει να γίνεται έλεγχος για το ποιος κέρδισε, καθώς και για το αν οι 2 παίκτες έχουν έρθει σε ισοπαλία.



- **Εφαρμογή Παραθυρικής Ηλεκτρονικής Σκακιάρας.** Βασικό επίπεδο υλοποίησης: Θα δημιουργήσετε μια παραθυρική εφαρμογή η οποία θα λειτουργεί ως σκακιάρα στην οποία μπορούν να παίξουν 2 παίχτες. Η εφαρμογή σας θα διαθέτει τα 8x8, 64 ασπρόμαυρα τετράγωνα που υπάρχουν στο σκάκι, καθώς και όλα τα διαθέσιμα πύονια. Ο κάθε παίχτης θα μπορεί να μετακινεί μέσα στην σκακιάρα τα πύονια του. Δεν χρειάζεται να γίνεται κάποιος έλεγχος από εσάς για τις μετακινήσεις, αυτό είναι ευθύνη των παικτών. Προχωρημένο επίπεδο υλοποίησης: Επιπλέον λειτουργίες: Ύπαρξη χρονομέτρου, τόσο για την ημερομηνία ώρα, όσο και για το πόσο χρόνο έχει ο κάθε παίχτης στη διάθεσή του. Με το που κάνει ένας παίχτης την κίνησή του, ο χρόνος του «παγώνει» και αντίστοιχα, αρχίζει η αντίστροφη μέτρηση του χρόνου του άλλου παίχτη. Η εφαρμογή σας θα πρέπει **να διαθέτει και ΒΔ**, στην οποία θα καταγράφονται τα παιχνίδια (ποιος παίχτης έχει παίξει με ποιον, καθώς και η χρονική στιγμή του παιχνιδιού).



- **Εφαρμογή Music Playlist.** Βασικό επίπεδο υλοποίησης: Θα υλοποιήσετε μια παραθυρική εφαρμογή αναπαραγωγής μουσικών κομματιών. Από μια λίστα επιλογής ανάμεσα στα διαθέσιμα μουσικά αρχεία, ο χρήστης θα μπορεί να επιλέγει το κομμάτι που θα παιχτεί, καθώς να έχει δυνατότητες παύσης και διακοπής του κομματιού. Για κάθε μουσικό κομμάτι που είναι αποθηκευμένο στην εφαρμογή σας θα υπάρχουν επίσης καταχωρημένα στοιχεία που θα συνοδεύουν το συγκεκριμένο κομμάτι, όπως «Όνομα καλλιτέχνη», «Έτος κυκλοφορίας», «Είδος Μουσικής», «Γλώσσα». Το πώς θα αποθηκευτούν τα δεδομένα αυτά αφήνεται ως δική σας επιλογή (π.χ. Χρήση απλών αρχείων κειμένου, ωστόσο καλύτερη λύση είναι η χρήση ΒΔ) Προχωρημένο επίπεδο υλοποίησης: Επιπλέον λειτουργίες: Καταγραφή των «πιο αγαπημένων» μουσικών κομματιών του χρήστη βάσει συχνότητας χρήσης τους. Επιλογή τυχαίας αναπαραγωγής κομματιών (random selection). Δυνατότητες διαμόρφωσης της λίστας των μουσικών κομματιών (προσθήκη νέου μουσικού κομματιού, διαγραφή μουσικού κομματιού, ενημέρωση στοιχείων υπάρχοντος μουσικού κομματιού).



\*Οι παραπάνω εικόνες είναι ενδεικτικές, δεν σημαίνει ότι πρέπει να κάνετε κάτι αντίστοιχο με αυτό που βλέπετε

### Σημαντικά (SOS - ΔΙΑΒΑΣΤΕ ΠΡΟΣΕΚΤΙΚΑ ΟΛΑ ΤΑ ΠΑΡΑΚΑΤΩ):

- Για να εξεταστεί μια ομάδα θα πρέπει να έχει δηλωθεί στον διδάσκοντα και να έχουν καταγραφεί τα μέλη της. Μη δηλωμένες ομάδες, δεν θα μπορούν να εξεταστούν σε καμία εξεταστική περίοδο.
- Η εξέταση των εργασιών θα γίνει μέσω MS Teams σε ημερομηνία/ώρα που θα ανακοινωθεί και θα είναι αμέσως μετά τη λήξη της εξεταστικής του Φεβρουαρίου. Η παρούσα τελική εργασία παραμένει ίδια και για την εξεταστική του Σεπτεμβρίου.
- Σε κάθε ομάδα είναι απαραίτητο **όλοι** οι φοιτητές να γνωρίζουν **όλες** τις λεπτομέρειες υλοποίησης και των 3 εφαρμογών. Κατά την εξέταση, εκτός από την επίδειξη των εργασιών στον υπολογιστή, **γίνεται και ταυτόχρονη προφορική** εξέταση, όπου οι εξεταστές ζητούν από κάθε φοιτητή ξεχωριστά να απαντήσει σε συγκεκριμένες ερωτήσεις, οι οποίες πολλές φορές αφορούν τον κώδικα των εφαρμογών, αλλά και εφ'όλης της ύλης γνώσεις προγραμματισμού σχετικά με τη C#. Λανθασμένες απαντήσεις αφαιρούν μονάδες από το συνολικό βαθμό των εργασιών, για τον/τους φοιτητή/φοιτητές που έκαναν τα λάθη.
- Η χρήση της C# ως γλώσσας προγραμματισμού είναι υποχρεωτική.
- Το εργαλείο ανάπτυξης των εφαρμογών είναι της επιλογής σας. Όμως συνιστάται η χρήση κάποιας έκδοσης Visual Studio, όσο πιο νέας και πληρέστερης γίνεται (υπενθυμίζεται ότι όλο το απαραίτητο software είναι διαθέσιμο στους φοιτητές).
- Για κάθε εφαρμογή έχουν οριστεί 2 επίπεδα υλοποίησης. Το 1<sup>ο</sup> (βασικό), εφόσον υλοποιηθεί, δίνει τη δυνατότητα σε μια ομάδα να πιάσει τη βάση για την εκάστοτε εφαρμογή. Το 2<sup>ο</sup> (προχωρημένο) δίδει στις ομάδες που το έχουν ολοκληρώσει το 100% του βαθμού.
- Για να βαθμολογηθεί μια εργασία με άριστα το 6 πρέπει να έχει υλοποιημένες και τις 3 εφαρμογές. Ωστόσο, μια ομάδα μπορεί να πάρει προβιβάσιμο βαθμό ακόμα και με 2 (καλά υλοποιημένες) από τις 3 εφαρμογές. Φυσικά συνυπολογίζονται και οι ατομικές εργασίες του κάθε φοιτητή.

- Εντός της κάθε ομάδας τα μέλη καλούνται να συνεργαστούν και να αλληλοβοηθηθούν. Ακόμα και στις περιπτώσεις που μεταξύ των ομάδων γίνει καταμερισμός των εργασιών (ποιος φοιτητής θα υλοποιήσει τι), θα πρέπει σε δεύτερο χρόνο να συναντηθείτε μεταξύ σας και να συζητήσετε τις λεπτομέρειες της υλοποίησης. **Δεν επιτρέπεται** η συνεργασία με άλλες ομάδες. Όμοιες εργασίες διαφορετικών ομάδων μηδενίζονται αυτόματα.
- **ΣΗΜΑΝΤΙΚΟ:** Κάθε εφαρμογή πρέπει να υλοποιηθεί με γνώμονα τον αντικειμενοστρεφή προγραμματισμό. Οι φοιτητές καλούνται να υλοποιήσουν όσο το δυνατόν περισσότερες τάξεις (όπου είναι αυτό εφικτό), οι οποίες να περιλαμβάνουν λειτουργίες και ιδιότητες. Σε κάθε εφαρμογή πρέπει να έχουν υλοποιηθεί **τουλάχιστον 2** κλάσεις (και φυσικά και αντικείμενα αυτών) δημιουργημένες από τους φοιτητές –και όχι αυτόματα από το Visual Studio (π.χ. Form Class)–, ακόμα και αν αυτές χρησιμοποιούν ή καλούν τμήματα άλλων τάξεων και οι οποίες να χρησιμοποιούνται σε κάποιο σημείο του κώδικα. Η χρήση «χαρακτηριστικών» του αντικειμενοστρεφούς προγραμματισμού (π.χ. κληρονομικότητα μεταξύ τάξεων) στις δικές σας τάξεις δεν είναι υποχρεωτική, ωστόσο θα εκτιμηθεί ιδιαίτερα εφόσον υλοποιηθεί από εσάς.
- **Απαγορεύεται αυστηρά** η αντιγραφή κώδικα είτε από το διαδίκτυο είτε από συμφοιτητές σας. Ο κώδικας όλων των εφαρμογών θα περάσει από ειδικό λογισμικό εντοπισμού λογοκλοπής κώδικα και στις περιπτώσεις αντιγραφής, οι εν λόγω εργασίες θα μηδενιστούν.
- Τυχόν απορίες που προκύπτουν κατά την εκπόνηση των εργασιών μπορούν να συζητηθούν **ΜΟΝΟ** κατά τη διάρκεια των μαθημάτων ή/και μέσω MS Teams και ΟΧΙ με την αποστολή e-mail στον διδάσκοντα!

Καλή επιτυχία!!