

# Report – Minesweeper σε C++ με raylib

---

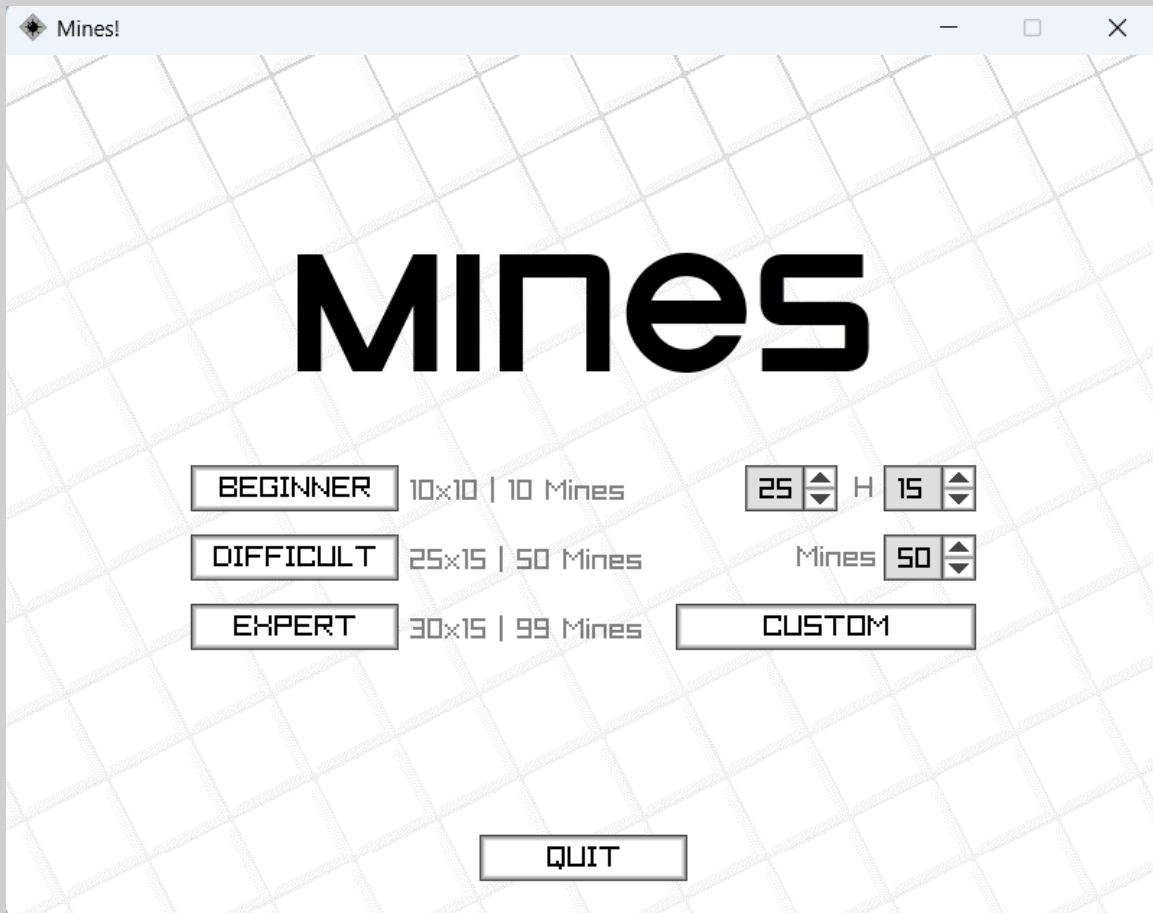
## Ονόματα Φοιτητών:

- Βράνας Γεώργιος 22135
- Τομάι Τζέσικα 22094
- Μαρτζούκος Κωνσταντίνος 22066

## Εισαγωγή

Το **Minesweeper** είναι ένα κλασικό παιχνίδι λογικής, όπου ο παίκτης πρέπει να αποκαλύψει όλα τα κελιά ενός πίνακα χωρίς να πέσει σε νάρκη. Ο στόχος της εργασίας ήταν να υλοποιηθεί το παιχνίδι σε C++ χρησιμοποιώντας τη βιβλιοθήκη **raylib**, ώστε να υπάρχει γραφικό περιβάλλον και ήχος.

Το project οργανώθηκε σε διάφορα αρχεία και κλάσεις, έτσι ώστε ο κώδικας να είναι πιο καθαρός και κατανοητός. Κάθε αρχείο έχει τον δικό του ρόλο στη λειτουργία του παιχνιδιού.

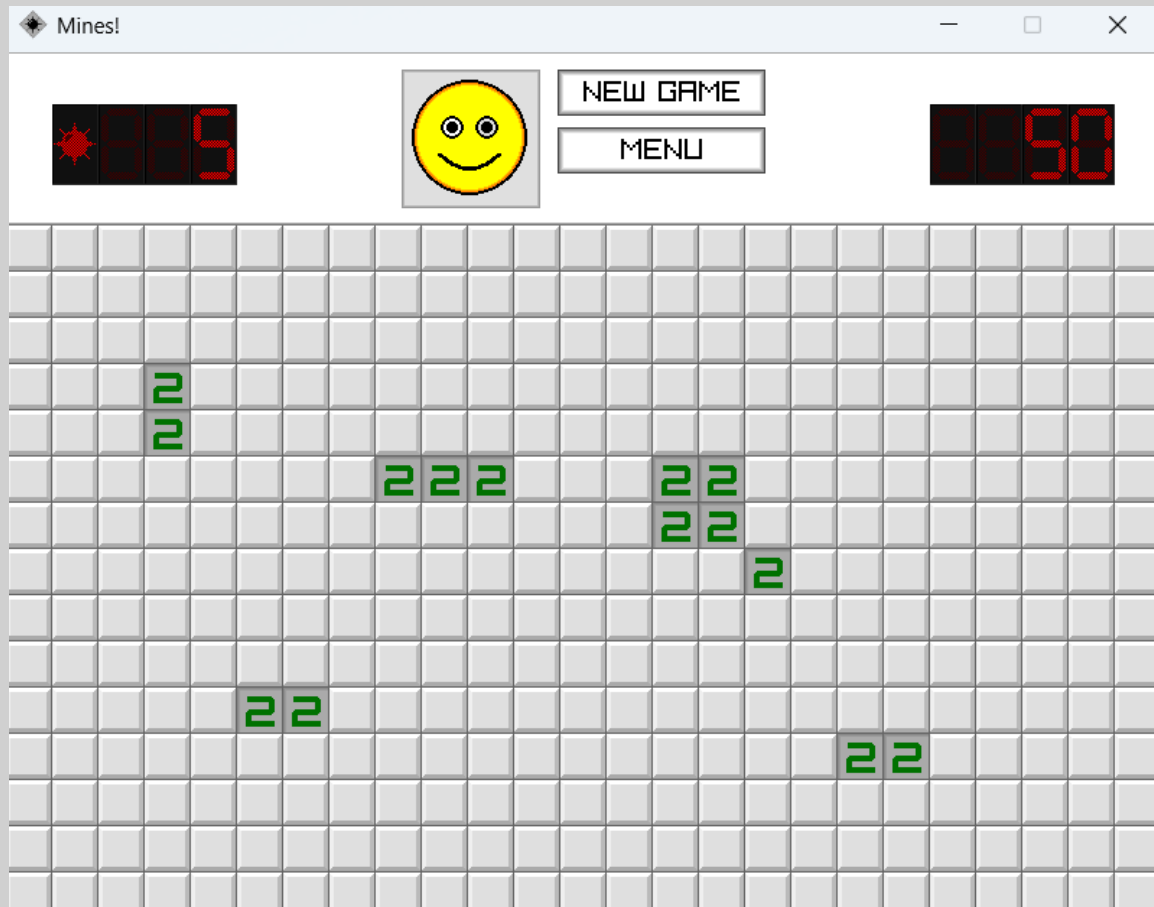


## game.cpp

Η κλάση `game.cpp` είναι ο βασικός κορμός του παιχνιδιού. Η κλάση `Game` διαχειρίζεται την αρχικοποίηση του παραθύρου και του ήχου, τον συντονισμό του μενού και του πίνακα με τις νάρκες, καθώς και τον τερματισμό του παιχνιδιού. Περιλαμβάνει επίσης μεθόδους για την έναρξη νέας παρτίδας, για την επανεκκίνηση και για την κατάσταση νίκης ή ήττας.

## game.h

Η κλάση `game.h` περιέχει τον ορισμό της κλάσης `Game`. Εκεί φαίνονται τα βασικά πεδία και οι συναρτήσεις που χρειάζονται για να λειτουργήσει το παιχνίδι, όπως η αρχικοποίηση, η ενημέρωση και η εκκίνηση μιας νέας παρτίδας. Στην ουσία είναι το «σχέδιο» της κλάσης.



## GUI.cpp

Η κλάση `GUI.cpp` υλοποιεί τα γραφικά στοιχεία της διεπαφής χρήστη. Εδώ υπάρχουν συναρτήσεις για κουμπιά που αλλάζουν όψη όταν ο χρήστης τα πατάει ή περνάει το ποντίκι από πάνω, καθώς και για αριθμητικά πεδία που επιτρέπουν την επιλογή τιμών. Με αυτόν τον τρόπο, το `GUI` κάνει την αλληλεπίδραση με το παιχνίδι πιο εύχρηστη και κατανοητή.

## **main.cpp**

Η κλάση **main.cpp** είναι το σημείο εκκίνησης του προγράμματος. Δημιουργεί ένα αντικείμενο **Game**, το αρχικοποιεί, τρέχει τον κύριο βρόχο του παιχνιδιού και στο τέλος το τερματίζει σωστά. Ουσιαστικά, λειτουργεί σαν ο «οδηγός» που ξεκινάει και σταματάει το παιχνίδι.

## **MenuScreen.cpp**

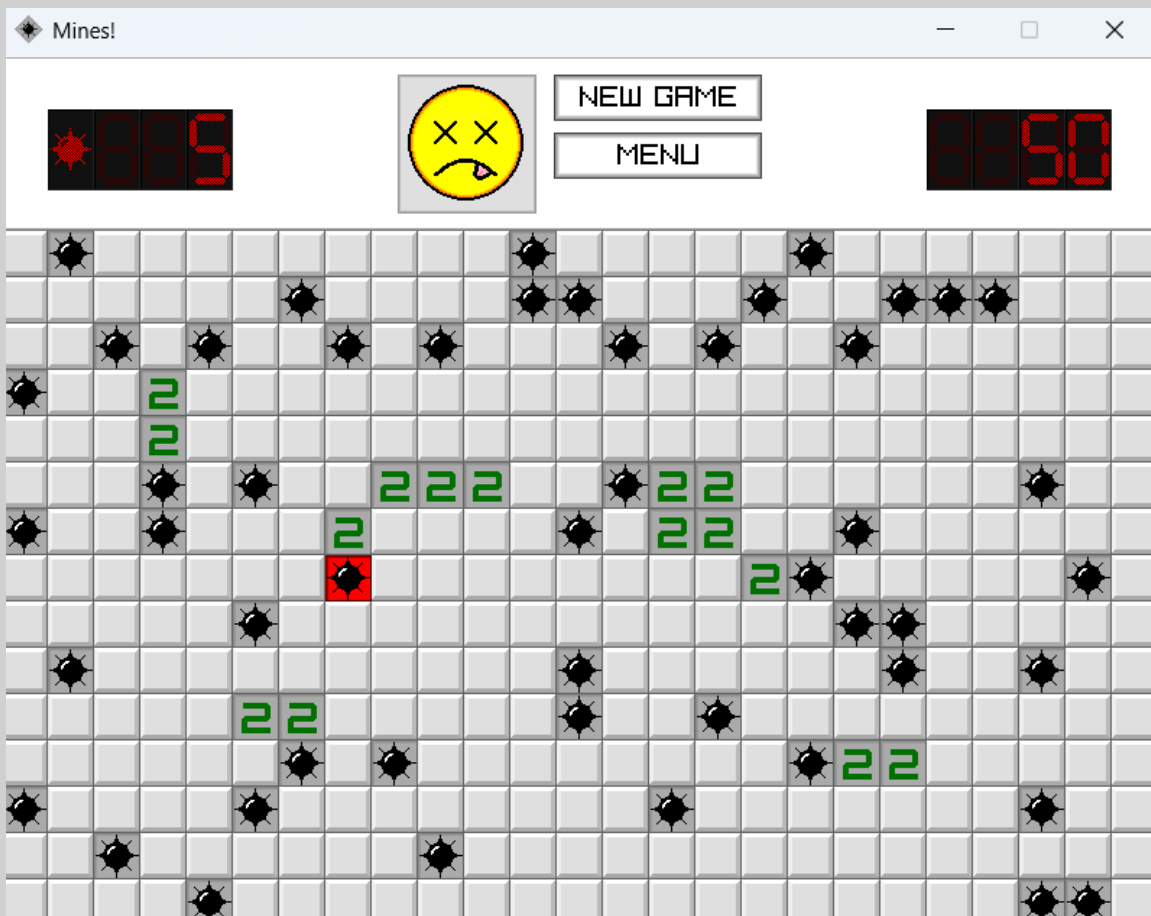
Η κλάση **MenuScreen.cpp** υλοποιεί την οθόνη μενού. Εκεί εμφανίζονται τα κουμπιά για επιλογή δυσκολίας (**Beginner**, **Difficult**, **Expert**) με διαφορετικά μεγέθη πίνακα και αριθμούς ναρκών. Υπάρχει και επιλογή «**Custom**», όπου ο παίκτης ορίζει μόνος του τις παραμέτρους. Το μενού περιλαμβάνει επίσης το κουμπί «**QUIT**» για τον τερματισμό.

## **Minefield.cpp**

Η κλάση **Minefield.cpp** είναι το πιο σημαντικό, καθώς περιέχει όλη τη λογική του παιχνιδιού. Η κλάση **Minefield** φροντίζει να δημιουργήσει τον πίνακα, να τοποθετήσει τις νάρκες τυχαία, να υπολογίσει τους αριθμούς γύρω από αυτές και να διαχειριστεί τις κινήσεις του παίκτη.

- Με την **ProcessGame()** ελέγχεται η είσοδος του χρήστη (αριστερό κλικ για άνοιγμα, δεξί κλικ για σημαία).
- Με την **Generate()** δημιουργείται ένας νέος πίνακας με νάρκες.
- Η **RevealEmptyCells()** ανοίγει αυτόματα τα κενά κελιά όταν δεν υπάρχουν νάρκες τριγύρω.
- Υπάρχει επίσης έλεγχος για το πότε κερδίζει ή χάνει ο παίκτης.

Το **Minefield** είναι το βασικό κομμάτι που καθορίζει την εμπειρία του παιχνιδιού.



### ResourceManager.cpp

Το αρχείο **ResourceManager.cpp** έχει ως κύρια αποστολή τη διαχείριση των πόρων που χρειάζεται το παιχνίδι για να λειτουργήσει σωστά. Στη μέθοδο **Init()** φορτώνονται όλα τα απαραίτητα αρχεία, όπως οι εικόνες για τα γραφικά στοιχεία (φόντο, κελιά, μπάρα κατάστασης, κουμπιά) αλλά και οι ήχοι για την έκρηξη και τη νίκη.

Αντίστοιχα, η μέθοδος **Shutdown()** φροντίζει να αποδεσμεύονται αυτοί οι πόροι όταν το παιχνίδι κλείνει, ώστε να μη μένουν άχρηστα δεδομένα στη μνήμη. Με αυτόν τον τρόπο, η κλάση **ResourceManager** λειτουργεί σαν «κεντρικός διαχειριστής» των αρχείων εικόνας και ήχου, εξασφαλίζοντας ότι όλα φορτώνονται σωστά στην αρχή και κλείνουν σωστά στο τέλος.

## Συμπεράσματα – Αποτέλεσμα Project

Η υλοποίηση του Minesweeper σε C++ με raylib έδειξε πώς μπορεί να οργανωθεί ένα παιχνίδι σε ξεχωριστές κλάσεις και αρχεία. Το αποτέλεσμα είναι ένα λειτουργικό παιχνίδι, όπου ο παίκτης μπορεί να επιλέξει επίπεδο δυσκολίας ή να ορίσει δικές του παραμέτρους. Υπάρχει μετρητής για τις σημαίες και διαφορετικές οπτικές ενδείξεις ανάλογα με το αν η παρτίδα κερδηθεί ή χαθεί.

Συνολικά, το project πέτυχε τον στόχο του να φτιαχτεί ένα πλήρες παιχνίδι με γραφικό περιβάλλον, εφαρμόζοντας αντικειμενοστραφή προγραμματισμό και κάνοντας πρακτική χρήση της βιβλιοθήκης raylib.