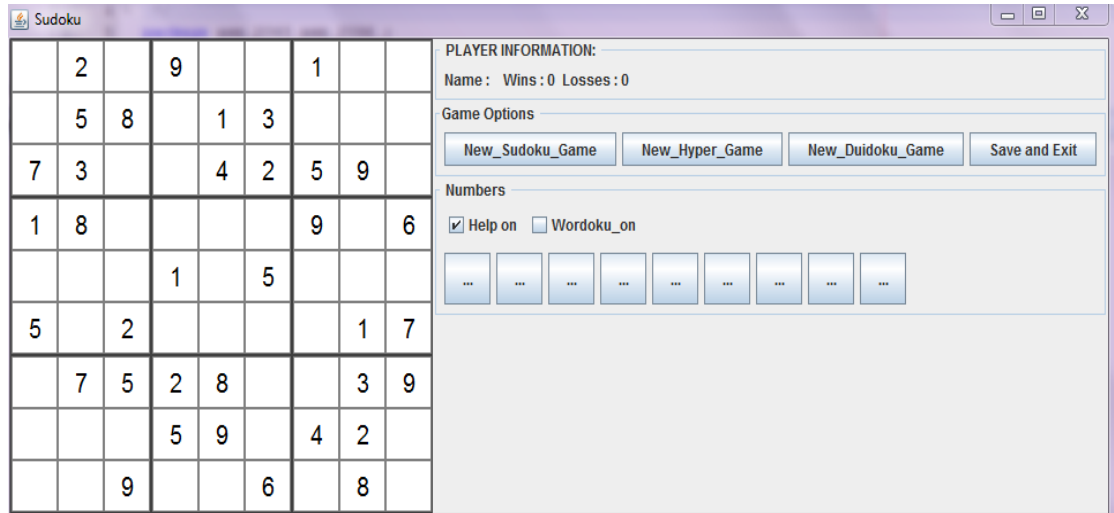


Φασούλας Θεόδωρος  
ΑΕΜ:2096  
email: [theodoft@csd.auth.gr](mailto:theodoft@csd.auth.gr)

Γεωργιάδης Νικόλαος  
ΑΕΜ:2043  
email: [nsgeorgi@csd.auth.gr](mailto:nsgeorgi@csd.auth.gr)

## Η ΕΦΑΡΜΟΓΗ SUDOKU ΣΕ JAVA

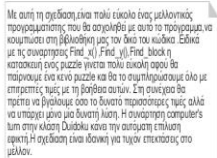


## ΕΙΣΑΓΩΓΗ

Η εφαρμογή αυτή μας δίνει τη δυνατότητα να παίξουμε το αγαπημένο μας παιχνίδι sudoku. Το παιχνίδι αυτό έχει 3 παραλλαγές : Classic sudoku, Hypersudoku, Duidoku.

Επίσης έχουμε την επιλογή της βοήθειας και της μετατροπής σε γράμματα. Η πρώτη αν ενεργοποιηθεί μας βοηθάει να συμπληρώσουμε το παζλ δειχνοντάς μας με μπλε κουτάκια που μπορούμε να εκχωρήσουμε τον εκάστοτε αριθμό. Η δεύτερη επιλογή αν ενεργοποιηθεί, μετατρέπει τους αριθμούς στα αντίστοιχα γράμματα. Όπως βλέπουμε η εφαρμογή κρατάει και διάφορες πληροφορίες για τον παίκτη όπως το ονομά του τις νίκες του και τις ήττες του. Ακόμα υπάρχει η επιλογή Save όπου ο παίκτης πριν τερματίσει τη λειτουργία του παιχνιδιού, αν δεν έχει ολοκληρώσει το παιχνίδι του μπορεί να το αποθηκεύσει έτσι ώστε την επόμενη φορά που θα ξεκινήσει τη διεπαφή να συνεχίσει από εκεί που βρισκόταν. Τέλος υπάρχουν τα 9 κουμπια τα οποία εκπροσωπούν τον κάθε αριθμο. **Για κάποιο λόγο δεν μου εμφάνιζε το νούμερο στο κάθε κουμπί ενώ λειτουργεί κανονικά.**

## ΥΛΟΠΟΙΗΣΗ ΣΕ JAVA



Το παραπάνω είναι το διάγραμμα κλάσεων της εφαρμογής μας. Το πρόγραμμα ξεκινάει από την κλάση Aem\_2043\_aem\_2096\_ όπου δημιουργούμε ένα αντικείμενο της κλάσης Gui. Η κλάση αυτή περιέχει την εφαρμογή δηλαδή τα γραφικά της και τη λογική της. Στον κατασκευαστή της κλάσης αυτής διαβάζουμε τα αρχεία από τον δίσκο δηλαδή το αρχείο test όπου έχει αποθηκευμένα 10 παιχνίδια για κάθε παραλλαγή, το αρχείο saved\_games όπου υπάρχει αποθηκευμένο το παιχνίδι της τελευταίας εκτέλεσης της εφαρμογής, και το αρχείο saved\_profiles όπου έχει τις πληροφορίες για κάθε παίκτη όπου έχει καταχώρηση το όνομά του στην εφαρμογή. Στη συνέχεια δημιουργούμε τόσο τα αντικείμενα της λογικής όσο και τα αντικείμενα για τα γραφικά. Η κλάση της λογικής είναι η Game η οποία περιέχει τα αντικείμενα της λογικής και των 3 παραλλαγών. Έτσι δημιουργούμε τα αντικείμενα αυτά και τα παιχνίδια από το αρχείο τα περνάμε σε κάθε αντικείμενο ξεχωριστά. Για τα αντικείμενα των γραφικών, το sudoku\_panel δημιουργεί το πάνελ για το Sudoku ενώ το button\_panel δημιουργεί το πάνελ για τα κουμπιά της εφαρμογής. Τα αντικείμενα αυτά επεκτείνουν την κλάση Observer για να ενημερώνουν ανά πάσα στιγμή τα πάνελ της εφαρμογής καθώς η αλληλεπίδραση του χρήστη με αυτά είναι

συνεχής. Το αντικείμενο `Sudoku_control` αναπαριστά την αλληλεπίδραση του χρήστη με το πάνελ του puzzle. Δηλαδή επεκτείνει την κλάση `MouseListener` που σημαίνει ότι σε οποιοδήποτε κλικ (δεξί ή αριστερό) του χρήστη πάνω στο puzzle, θα κληθεί η συνάρτηση `mousepressed()` και ανάλογα με το κελί που πάτησε, θα εκτελούνται διάφορες συναρτήσεις. Το αντικείμενο `button_control` αναπαριστά την αλληλεπίδραση του χρήστη με το πάνελ των κουμπιών. Δηλαδή επεκτείνει την κλάση `ActionListener` που σημαίνει ότι σε οποιαδήποτε κλικ του χρήστη (μόνο αριστερό) πάνω στο πάνελ αυτό θα κληθεί η συνάρτηση `actionPerformed(ActionEvent e)` και ανάλογα με το κουμπί που πάτησε θα εκτελούνται διάφορες συναρτήσεις.

Εάν δεν υπάρχει προηγούμενο αποθηκευμένο παιχνίδι, τότε η διεπαφή (αφού ρωτήσει τον χρήστη εάν είναι παλιός ή καινούργιος παίκτης για να του φορτώσει τα στατιστικά του.) αρχίζει με ένα παιχνίδι της κλασσικής έκδοσης. Αυτό γίνεται καλώντας την συνάρτηση `New_Game()` όπου μέσα σε αυτήν επιλέγοντας τυχαία ένα από τα 10 παζλ της έκδοσης αυτής το αποθηκεύει στο πίνακα `current_game`. Διαφορετικά στον πίνακα αυτόν θα εκχωρηθεί το `saved_game`. Για να ενημερωθούν όμως τα γραφικά και να δούμε το παζλ στην οθόνη μας θα πρέπει να καλέσουμε τις συναρτήσεις `setChanged()`; `notifyObservers(Update_Action.NEW_GAME_SUDOKU)`; Η κλάση `Update` είναι enumeration και χρησιμοποιείται για να ενημερώνει τους παρατηρητές κάθε φορά, τη είδους ενημέρωση κάνει. Η `NEW_GAME_SUDOKU` καλεί τη `setGame((Classic_Sudoku_Game) o)`; Όπου ενημερώνεται το κάθε κελί `field`. Η κλάση `Field` αναπαριστά το κάθε κελί του παζλ.

Ετσι αφού έχει φορτώσει η εφαρμογή η αλληλεπίδραση του χρήστη με αυτή θα συνεχίσει την εκτέλεση διάφορων συναρτήσεων που έχουμε υλοποιήσει.

## **BUTTON NEW GAME SUDOKU GAME**

## **BUTTON NEW HYPER SUDOKU GAME**

## **BUTTON NEW DUIDOKU GAME**

Την χρήση των παραπάνω κουμπιών την εξηγήσαμε παραπάνω. Μόλις πατηθεί ένα από αυτά τα κουμπία θα κληθεί η συνάρτηση `New_Game()` (η οποία υπάρχει για την εκάστοτε παραλλαγή αφού έχει γίνει `Override` από την κλάση `Sudoku`) και αυτή με την σειρά της θα καλέσει την `setGame`.

## **CHECK BOX HELP ON**

Όταν πατηθεί αυτό το κουμπί θα κληθεί η `setHelp()` η οποία με τη σειρά της θα καλέσει την `setCandidates()` όπου ελέγχουμε εάν το help button είναι ενεργο και αν `isSelectedNumberCandidate(x, y)` είναι true. Η τελευταία ελέγχει εάν ο αριθμός που επέλεξε ο χρήστης είναι εγκυρος. Δηλαδή είναι ο μοναδικός αριθμός σε μια συγκεκριμένη στήλη, γραμμή και στο block το οποίο ανήκει. Εάν ισχύουν οι

παραπάνω συνθήκες τότε θα εμφανιστούν μπλε κουτάκια στο πάνελ του παζλ που θα δείχνουν σε ποιες θέσεις μπορεί ο χρήστης να βάλει τον αριθμό.

## **CHECK BOX WORDOKU ON**

Όταν πατηθεί αυτό το κουμπί θα κληθεί η `convert_letter_to_number()` η οποία με τη σειρά της θα καλέσει την `setGame ((Classic_Sudoku_Game) o)`; Εάν το το wordoku check box είναι ενεργό τότε θα καλέσει η συνάρτηση `setNumber_withletters ()` η οποία μετατρέπει τους αριθμούς σε γράμματα (`char`)( `number + 64`)).

## **BUTTON NUMBERS**

Όταν πατηθεί ένα κουμπί από τους αριθμούς τότε ενεργοποιείται η συνάρτηση `setSelectedNumber(Integer.parseInt(e.getActionCommand()))` η οποία αποθηκεύει στη μεταβλητή `SelectedNumber` τον αριθμό που επέλεξε ο χρήστης .Μετά `setCandidates()` όπου εξηγήσαμε την χρήση της παραπάνω.

## **BUTTON SAVE AND EXIT**

Όταν πατηθεί αυτό το κουμπί θα κληθεί θα αποθηκεύσουμε σε ένα αρχείο `saved_game` το τρέχων παιχνίδι του χρήστη καθώς και στο αρχείο `saved_profiles` θα αποθηκεύσουμε τα στατιστικά μέχρι στιγμής του χρήστη.Στη συνέχεια θα τερματίσει η εφαρμογή.

## **MOUSE PRESSED ΣΤΟ ΠΑΝΕΛ ΤΟΥ ΠΑΖΛ**

Όταν πατηθεί αριστερό κλικ σε ένα κελί του παζλ τότε ενεργοποιείται η `mousePressed(MouseEvent e)` όπου γίνεται μια σειρά από ελέγχους. Αν το κελί δεν περιέχει αριθμό και αν ο αριθμός που επέλεξε ο χρήστης είναι εγκυρος μέσω της συνάρτησης `isSelectedNumberCandidate(x, y)` τότε ο αριθμός εκχωρείται.Αν ο χρήστης πατήσει δεξί κλικ τότε ο αριθμός που έδωσε τελευταία , εξαφανίζεται από το παζλ εφόσον είναι με μπλε χρώμα ο αριθμός που σημαίνει ότι ήταν ο τελευταίος

.

## **DUIDOKU**

Στην παραλλαγή αυτή αξίζει να δούμε τη χρήση της συνάρτησης `Find_new_block_duidoku` όπου ελέγχει το block στο οποίο βρίσκεται ο αριθμός που επέλεξε ο χρήστης έτσι ώστε η τιμή αυτή να είναι μοναδική.Για να βρούμε σε ποιο block πρέπει να είναι εγκυρος ο αριθμός αρχικοποιούμε την συντεταγμένη `x` με 4 εάν είναι μικρότερη του 6 , αλλιώς με 6 . Για την συντεταγμένη `y` εάν είναι μικρότερη του 3 με 1 αλλιώς με 3.Στη συνέχεια για loop από αυτές τις αρχικές τιμές +1 εξετάζουμε εάν ο αριθμός είναι μοναδικός.

## ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ