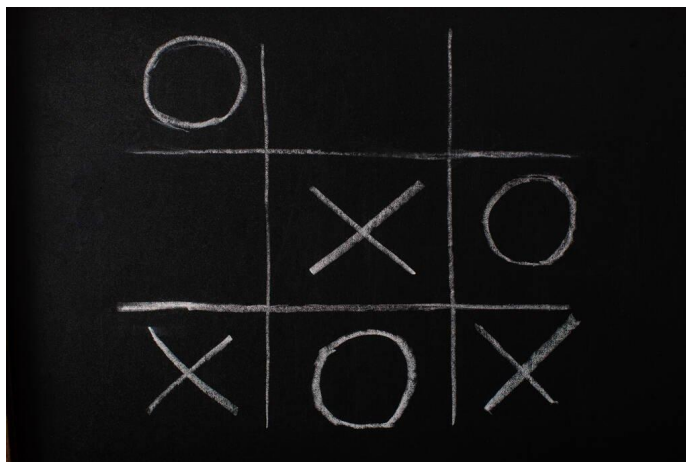


Τρίλιζα (Tic Tac Toe)

1. Εισαγωγή



Εικόνα 1: Τρίλιζα.

- Περιγραφή Παιχνιδιού:

Η Τρίλιζα (Tic Tac Toe) είναι ένα επιτραπέζιο παιχνίδι, το οποίο παίζεται σε ένα πλέγμα 3 προς 3 από δύο παίκτες, οι οποίοι τοποθετούν εναλλάξ τα σημάδια "X" και "O" σε ένα από τα εννέα κενά του πλέγματος.

Δεν υπάρχει κανόνας για το ποιος παίζει πρώτος, ωστόσο έχει τεθεί σύμβαση να ξεκινάει το παιχνίδι ο "X". Ο παίκτης που καταφέρνει να τοποθετήσει τρία από τα σημάδια του σε μια οριζόντια, κάθετη ή διαγώνια σειρά του πλέγματος είναι ο νικητής. Στην περίπτωση που κανένας από τους δύο παίκτες δεν καταφέρει να σχηματίσει τριάδα, προκύπτει ισοπαλία.

- Πρακτική Εφαρμογή:

Λόγω της απλότητας του παιχνιδιού, το Tic Tac Toe χρησιμοποιείται συχνά ως ένα παιδαγωγικό εργαλείο για τη διδασκαλία εννοιών όπως:

της ηθικής του «Ευ Αγωνίζεστε», της ανάπτυξης δεξιοτήτων τακτικής, αλλά και της καλλιέργειας του νου και της σκέψης.

- Ιστορική Εξέλιξη:

Η προέλευση του Tic Tac Toe αποδίδεται στους αρχαίους Αιγύπτιους, όπου και εμφανίστηκε περίπου το 1300 π.Χ. Όμως, η πιο αρχαία αναφορά που έχουμε για

το παιχνίδι είναι από τους Ρωμαίους, το οποίο αναφέρονταν ως Terni Lapilli (τρία βότσαλα τη φορά).

Το 1952, ο Βρετανός επιστήμονας υπολογιστών Sandy Douglas ανέπτυξε το Tic Tac Toe για τον υπολογιστή EDSAC στο Πανεπιστήμιο του Cambridge, το οποίο και έγινε ένα από τα πρώτα γνωστά βιντεοπαιχνίδια. Ο υπολογιστής μπορούσε να παίζει κανονικά, Τρίλιζα, εναντίον ενός ανθρώπινου αντιπάλου.

Το 1975, το Tic Tac Toe χρησιμοποιήθηκε επίσης και από φοιτητές του MIT για να επιδείξουν την υπολογιστική ισχύ των στοιχείων Tinkertoy. Ο υπολογιστής Tinkertoy, κατασκευασμένος από (σχεδόν) μόνο Tinkertoys, μπορεί να παίζει τέλεια Τρίλιζα. Αυτή τη στιγμή εκτίθεται στο Μουσείο Επιστημών της Βοστώνης.

- **Αλγόριθμος Νίκης - Victory Algorithm:**

Σύμφωνα με το πρόγραμμα Tic Tac Toe των Newell και Simon's (1972) και βάση του αλγορίθμου τους, ένας παίκτης μπορεί να σχηματίσει την «Τέλεια Τρίλιζα», για να κερδίσει ή τουλάχιστον να ισοφαρίσει το παιχνίδι, εάν κάθε φορά που είναι η σειρά του να παίζει, επιλέξει ως κίνηση μία από τις εξής περιπτώσεις:

Win: Εάν ο παίκτης έχει δύο σημάδια στη σειρά, μπορεί να τοποθετήσει ένα τρίτο για να σχηματιστούν τρία και να νικήσει.

Block: Εάν ο αντίπαλος έχει δύο σημάδια στη σειρά, ο παίκτης πρέπει να παίζει το τρίτο στη σειρά για να μπλοκάρει τον αντίπαλο.

Fork: Προκαλείται ένα σενάριο στο οποίο ο παίκτης έχει δύο τρόπους να κερδίσει (Δύο μη μπλοκαρισμένες γραμμές των 2 σημαδιών).

Blocking an opponent's fork: Εάν υπάρχει μόνο ένα δυνατό Fork για τον αντίπαλο, ο παίκτης θα πρέπει να το μπλοκάρει. Διαφορετικά, ο παίκτης θα πρέπει να μπλοκάρει όλα τα Forks με οποιονδήποτε τρόπο που να του επιτρέπει ταυτόχρονα να κάνει δύο στη σειρά. Σε κάθε άλλη περίπτωση, ο παίκτης θα πρέπει να κάνει δύο στη σειρά για να αναγκάσει τον αντίπαλο να αμυνθεί, αρκεί να μην έχει ως αποτέλεσμα να παράγει ένα Fork. Για παράδειγμα, εάν ο "X" έχει δύο αντίθετες γωνίες και ο "O" το κέντρο, ο "O" δεν πρέπει να παίζει στην γωνία για να κερδίσει. (Παίζοντας μια κίνηση στην γωνία σε αυτό το σενάριο δημιουργείται ένα Fork για να κερδίσει ο "X".)

Center: Ένας παίκτης μαρκάρει το κέντρο. Αν είναι η πρώτη κίνηση του παιχνιδιού, το να παίζετε μια κίνηση στην γωνία δίνει στον δεύτερο παίκτη περισσότερες ευκαιρίες να κάνει λάθος και επομένως μπορεί να είναι η καλύτερη επιλογή σας. Ωστόσο, δεν έχει καμία διαφορά μεταξύ των τέλειων παικτών.

Opposite Corner: Εάν ο αντίπαλος βρίσκεται στη γωνία, ο παίκτης παίζει στην απέναντι γωνία.

Empty Corner: Ο παίκτης παίζει σε γωνιακό τετράγωνο.

Empty Side: Ο παίκτης παίζει σε ένα μεσαίο τετράγωνο σε οποιαδήποτε από τις τέσσερις πλευρές.

2. Σχεδίαση

Κύρια Εικόνα Navigator:

Δέντρο Σχεδίασης:

JFrame

□ Background Panel

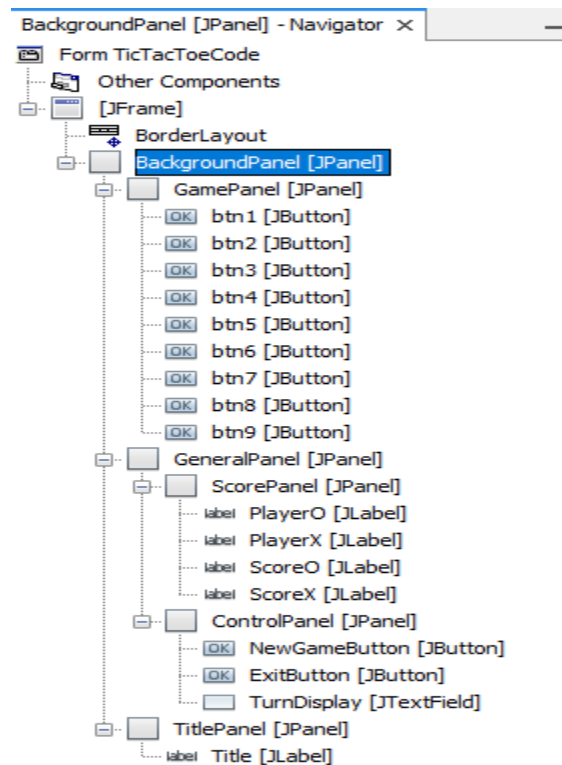
□ GamePanel

□ GeneralPanel

□ Score Panel

□ Control Panel

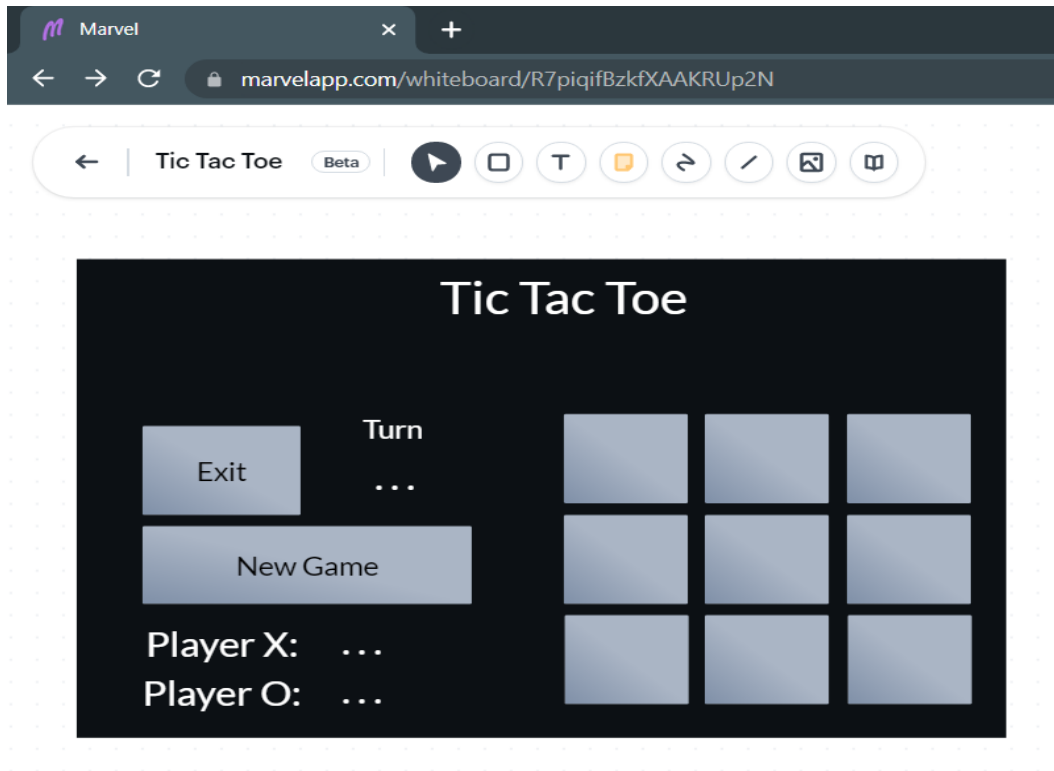
□ Title Panel



Εικόνα 2: “Tic Tac Toe” Navigator in NetBeans.

- Αναλυτικά Βήματα Σχεδίασης:

Η Σχεδίαση της εφαρμογής "Tic Tac Toe" και η τοποθέτηση κάθε αντικειμένου (π.χ JButton, JPanel κλπ.) βασίζεται στο παρακάτω πρωτότυπο (Εικόνα 3). Δημιουργήσαμε το πρωτότυπο αυτό μέσω της Πλατφόρμας/Εφαρμογής **Marvel** (<https://marvelapp.com/>). Η προβολή του πρωτοτύπου είναι διαθέσιμη και μέσω του συνδέσμου: (<https://marvelapp.com/whiteboard/R7piqifBzkfXAAKRU2N>).



Εικόνα 3: "Tic Tac Toe" Prototype made by Marvel App.

Αρχικά, στο NetBeans, δημιουργήσαμε το project "Tic Tac Toe" φτιάχνοντας ένα **JFrame Form** και αλλάζοντας τις εξής του Ιδιότητες (Properties):

JFrame:

Set Layout: Border Layout (Default)
defaultCloseOperation: EXIT_ON_CLOSE
title: Tic Tac Toe
preferredSize : [1091, 671]

Έπειτα δημιουργήσαμε ένα JPanel με όνομα **BackgroundPanel** με σκοπό να χρησιμοποιηθεί ως background της εφαρμογής. Προσαρμόσαμε σ' αυτό τις εξής Ιδιότητες (Properties):

BackgroundPanel:

Properties:

background: [0,0,0]
border: (No Border)
preferredSize: [1091, 671]
Layout □ Direction: Center

Responsive Design:

Set Layout: Free Design

Εσωτερικά στο BackgroundPanel:

Προσθέσαμε 3 JPanel με ονόματα **GamePanel**, **GeneralPanel** και **TitlePanel**, με τις εξής Ιδιότητες (Properties):

GamePanel:

Properties:

background: [0,0,0]
border: Bevel Border
preferredSize: [32, 32]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

Εσωτερικά έχουν τοποθετηθεί κατάλληλα 9 **JButton** με τέτοιο τρόπο ώστε να σχηματιστεί το περιβάλλον της Τρίλιζας, πάνω στο οποίο θα παίζουν οι χρήστες. Αυτά περιέχουν τις εξής Ιδιότητες (Properties) και Γεγονότα (Events):

JButton:

Properties:

background: [0,0,0]
font: Tempus Sans ITC 111 Plain
foreground: [0,0,0]
focusable: null
preferredSize: [200, 200]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

Events:

actionPerformed: btn1ActionPerformed

* Οι παραπάνω Ιδιότητες (Properties), Γεγονότα (Events) και το Responsiveness ισχύουν για κάθε ένα από τα 9 JButtons.

GeneralPanel:

Properties:

background: [0,0,0]
border: Bevel Border
foreground: [0,0,0]
focusable: true
preferredSize: [433, 482]

Responsive Design:

Horizontal Resizable: false
Vertical Resizable: true

Εσωτερικά στο GeneralPanel:

Προσθέσαμε 2 JPanel με ονόματα **ScorePanel** και **ControlPanel**, με τις εξής Ιδιότητες (Properties):

ScorePanel:

Properties:

background: [0,0,0]
border: Bevel Border
foreground: [0,0,0]
focusable: true
preferredSize: [326, 189]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

Εσωτερικά στο ScorePanel:

Προσθέσαμε 4 JLabel με ονόματα **PlayerO**, **PlayerX**, **ScoreO** και **ScoreX**, με τις εξής Ιδιότητες (Properties):

PlayerO:

Properties:

background: [240,240,240]
font: Tempus Sans ITC 66 Plain
text: Player O:
foreground: [255,0,255]
horizontalAlignment: LEADING
verticalAlignment: CENTER
verticalTextPosition: CENTER
focusable: true
preferredSize: [254, 87]

Responsive Design:

Horizontal Resizable: true

Vertical Resizable: true

PlayerX:

Properties:

background: [240,240,240]
font: Tempus Sans ITC 66 Plain
text: Player X:
foreground: Green
horizontalAlignment: LEADING
verticalAlignment: CENTER
verticalTextPosition: CENTER
focusable: true
preferredSize: [244, 87]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

ScoreO:

Properties:

background: [240,240,240]
font: Tempus Sans ITC 66 Plain
text: 0
foreground: [255,255,0]
horizontalAlignment: CENTER
verticalAlignment: CENTER
verticalTextPosition: CENTER
focusable: true
preferredSize: [44, 87]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

ScoreX:

Properties:

background: [240,240,240]
font: Tempus Sans ITC 66 Plain
text: 0
foreground: [255,255,0]
horizontalAlignment: CENTER
verticalAlignment: CENTER
verticalTextPosition: CENTER
focusable: true
preferredSize: [44, 87]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

ControlPanel:

Properties:

background: [0,0,0]
border: Bevel Border
foreground: [0,0,0]
focusable: true
preferredSize: [383, 241]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

Εσωτερικά στο **ControlPanel:**

Προσθέσαμε 2 JButton με ονόματα **NewGameButton**, **ExitButton** και ένα JTextField με όνομα **TurnDisplay**, και στην συνέχεια εφαρμόσαμε τις εξής Ιδιότητες (Properties) και Γεγονότα (Events):

NewGameButton:

Properties:

background: [0,0,0]
font: Tempus Sans ITC 68 Bold
foreground: [0,255,255]
mnemonic: N
displayedMnemonicIndex: -1
focusable: null
text: New Game
preferredSize: [200, 200]
horizontalAlignment: CENTER
verticalAlignment: CENTER
verticalTextPosition: CENTER
horizontalTextPosition: TRAILING

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

Events:

actionPerformed: NewGameButtonActionPerformed

ExitButton:

Properties:

background: [0,0,0]
font: Tempus Sans ITC 70 Bold
foreground: Red
mnemonic: E
displayedMnemonicIndex: -1
focusable: null
text: Exit
preferredSize: [200, 200]
horizontalAlignment: CENTER

verticalAlignment: CENTER
verticalTextPosition: CENTER
horizontalTextPosition: TRAILING

Responsive Design:

Horizontal Resizable: false
Vertical Resizable: true

Events:

actionPerformed: ExitButtonActionPerformed

TurnDisplay:

Properties:

editable: null
background: [0,0,0]
font: Tempus Sans ITC 36 Bold
foreground: Green
focusable: true
text: Player X
border: TitleBorder

- ☐ Title: Turn
- ☐ Color: [255,255,0]
- ☐ Font: Tempus Sans ITC 36 Bold
- ☐ Justification: Center
- ☐ Position: Top

preferredSize: [200, 200]
horizontalAlignment: CENTER
verticalAlignment: CENTER
verticalTextPosition: CENTER
horizontalTextPosition: TRAILING

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

Events:

actionPerformed: TurnDisplayActionPerformed

Εσωτερικά στο **BackgroundPanel**:

Να σημειωθεί ξανά πως εκτός από τα **GamePanel** και **GeneralPanel**, που προσαρμόσαμε αναλυτικά παραπάνω, προσθέσαμε και ένα JPanel με όνομα **TitlePanel** όπου θα αναγράφεται ο τίτλος του παιχνιδιού. Το JPanel αυτό θα έχει τις εξής Ιδιότητες (Properties):

TitlePanel:

Properties:

background: [0,0,0]
border: Bevel Border
foreground: [0,0,0]
focusable: true
preferredSize: [1047, 90]

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: false

Εσωτερικά στο **TitlePanel**:

Προσθέσαμε ένα JLabel με όνομα **Title**, το οποίο περιλαμβάνει τις εξής Ιδιότητες (Properties):

Title:

Properties:

background: [0,0,0]
font: Tempus Sans ITC 70 Plain
foreground: [255,255,0]
focusable: null
text: Tic Tac Toe
preferredSize: [200, 200]
horizontalAlignment: CENTER

verticalAlignment: CENTER
verticalTextPosition: CENTER
horizontalTextPosition: TRAILING

Responsive Design:

Horizontal Resizable: true
Vertical Resizable: true

3. Παρουσίαση Εφαρμογής



Εικόνα 4: Running Application "Tic Tac Toe" in NetBeans.

Τρέχοντας την εφαρμογή, διακρίνονται 4 Panels, όπως φαίνονται στην *Εικόνα 4*.

- Το 1^ο στο πάνω μέρος του παραθύρου είναι το **TitlePanel** στο οποίο αναγράφεται ο τίτλος του παιχνιδιού "Tic Tac Toe".
- Το 2^ο στα αριστερά, κάτω από τον τίτλο, είναι το **ControlPanel** το οποίο περιέχει 2 Buttons και ένα Text Field. Το πρώτο Button είναι το **ExitButton** και σκοπός του είναι να τερματίσει την εφαρμογή όποτε το θελήσει ο χρήστης. Από κάτω του βρίσκεται το **NewGameButton** το οποίο δημιουργεί ένα νέο παιχνίδι, μηδενίζοντας τα score των δύο παικτών. Το Text Field **TurnDisplay** είναι υπεύθυνο να αναδεικνύει την τρέχουσα σειρά του παίκτη που θα παίζει Τρίλιζα.

- Το 3^ο, κάτω από το **ControlPanel**, είναι το **ScorePanel** όπου αναγράφονται τα score των παικτών Player X και Player O αντίστοιχα. Τα **ControlPanel** και **ScorePanel** μαζί καθυστερούν το **GeneralPanel** της εφαρμογής.
- Τέλος, το 4^ο ονομάζεται **GamePanel**. Βρίσκεται στην δεξιά πλευρά της εφαρμογής, η οποία είναι αφιερωμένη στην Τρίλιζα. Σε αυτό έχουν τοποθετηθεί 9 κατάλληλα Buttons τα οποία κάνουν Interface με τους χρήστες/παίκτες του παιχνιδιού.

4. Υλοποίηση Εφαρμογής

Ανάπτυξη Κώδικα

Μετά την ολοκλήρωση της Σχεδίασης της Εφαρμογής, ακολουθεί η Ανάπτυξη του Κώδικα. Αρχικά προγραμματίσαμε τα 2 JButton (δηλαδή τα **NewGameButton** και **ExitButton**), τα οποία βρίσκονται στο **GeneralPanel** και συγκεκριμένα εντός του **ControlPanel**.

```
private void NewGameButtonActionPerformed(java.awt.event.ActionEvent evt) {
    btn1.setEnabled(true);
    btn2.setEnabled(true);
    btn3.setEnabled(true);
    btn4.setEnabled(true);
    btn5.setEnabled(true);
    btn6.setEnabled(true);
    btn7.setEnabled(true);
    btn8.setEnabled(true);
    btn9.setEnabled(true);

    ScoreX.setText("0");
    ScoreO.setText("0");

    XCount = 0;
    OCount = 0;
    BCount = 0;

    btn1.setText("");
    btn2.setText("");
    btn3.setText("");
    btn4.setText("");
    btn5.setText("");
    btn6.setText("");
    btn7.setText("");
    btn8.setText("");
    btn9.setText("");
}
```

Εικόνα 5: NewGameButtonActionPerformed().

Με το πάτημα του **NewGameButton** δημιουργείται ένα Γεγονός (Event) όπου όλα τα Buttons της Τρίλιζας γίνονται διαθέσιμα, για χρήση, από τους παίκτες του παιχνιδιού. Παράλληλα, στο **ScorePanel**, μηδενίζονται τα σκόρ των 2 παικτών.

Συγκεκριμένα, δημιουργήσαμε/προγραμματίσαμε την Συνάρτηση **NewGameButtonActionPerformed()**, όπως φαίνεται στην *Εικόνα 5*, ως εξής:

- Ενεργοποιήσαμε κάθε ένα Button ξεχωριστά με την συνάρτηση `setEnabled()`; την οποία μας παρέχει η βιβλιοθήκη της JavaSwing. Π.χ `btn1.setEnabled(true)`;
- Αλλάξαμε τα Text των σκόρ (ScoreX και ScoreO) των 2 παικτών μέσω της συνάρτησης `setText()`; και τα μηδενίσαμε. Π.χ `ScoreX.setText("0")`;
- Έπειτα, μηδενίσαμε και τους 3 μετρητές XCount, OCount και BCount. Σημειώνεται ότι ο 1^{ος} μετρητής είναι για το σκορ του Player X, ο 2^{ος} για το σκορ του Player O και ο 3^{ος} για το σύνολο των Buttons που έχουν πατηθεί για την περίπτωση της ισοπαλίας.
- Τέλος, αλλάξαμε και τα Text όλων των 9 Buttons, μέσω της `setText()`; συνάρτησης, αφήνοντας το String τους κενό. Π.χ `btn1.setText("")`;

```
private void ExitButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    frame = new JFrame("Exit");  
    if(JOptionPane.showConfirmDialog(frame, "Are you sure you want to exit?", "Tic Tac Toe"  
    , JOptionPane.YES_NO_OPTION) == JOptionPane.YES_NO_OPTION) {  
        System.exit(0);  
    }  
}
```

Εικόνα 6: ExitButtonActionPerformed().

Με το πάτημα του **ExitButton** δημιουργείται ένα Γεγονός (Event) όπου τερματίζεται η εφαρμογή όποτε το θελήσει ο χρήστης. Με το κλικ αυτού του Button εμφανίζεται και ένα μικρό παράθυρο για την επιβεβαίωση της εξόδου της εφαρμογής, σε περίπτωση που ο χρήστης το πατήσει καταλάθος (missclick).

Συγκεκριμένα, προγραμματίσαμε την Συνάρτηση **ExitButtonActionPerformed()**, όπως φαίνεται στην *Εικόνα 6*, ως εξής:

- Δημιουργήσαμε ένα νέο JFrame με όνομα "Exit": `frame = new Jframe("Exit")`;
- Με μια δομή ελέγχου `if` δώσαμε την επιλογή εξόδου ή συνέχειας του παιχνιδιού στον χρήστη. Έτσι, μέσω ενός μικρού παραθύρου JFrame θα εμφανίζεται ενημερωτικό μήνυμα "Are you sure you want to exit? " με επιλογές YES or NO για τον χρήστη.
- Με την επιλογή YES (δηλαδή: εάν ισχύει η συνθήκη του `if`) διακόψαμε και τερματίσαμε το πρόγραμμα μέσω της εντολής `System.exit(0)`. Διαφορετικά, το παιχνίδι συνεχίζει να τρέχει.

```
private void gameScore() {
    ScoreX.setText(String.valueOf(XCount));
    ScoreO.setText(String.valueOf(OCount));
}
```

Εικόνα 7: gameScore().

Δημιουργήσαμε, επιπλέον, την Συνάρτηση **gameScore()**, με σκοπό να αναγράφονται τα σκόρ των 2 παικτών στα JLabel **ScoreX** και **ScoreO** αντίστοιχα. Ο κώδικας της Συνάρτησης αυτής φαίνεται, παραπάνω, στην *Εικόνα 7*.

```
private void ChooseAPlayer() {
    if(startGame.equalsIgnoreCase("X")) {
        startGame = "O";
    }
    else {
        startGame = "X";
    }
}
```

Εικόνα 8: ChooseAPlayer().

Στην συνέχεια, δημιουργήσαμε και μια Συνάρτηση με όνομα **ChooseAPlayer()**, έτσι ώστε να εναλλάσσεται η σειρά του παίκτη που παίζει πρώτος. Έτσι, μέσω μίας απλής δομής ελέγχου if, όπως παρατηρείτε στην *Εικόνα 8*, θα γίνεται η εναλλαγή.

```
public class TicTacToeCode extends javax.swing.JFrame {

    /**
     * Creates new form TicTacToeCode
     */
    private String startGame = "X";
    private int XCount = 0;
    private int OCount = 0;

    public TicTacToeCode() {
        initComponents();
    }

    private void gameScore() {
        ScoreX.setText(String.valueOf(XCount));
        ScoreO.setText(String.valueOf(OCount));
    }

    private void ChooseAPlayer() {
        if(startGame.equalsIgnoreCase("X")) {
            startGame = "O";
        }
        else {
            startGame = "X";
        }
    }
}
```

Εικόνα 9: Γενική Εικόνα Συναρτήσεων gameScore() και ChooseAPlayer().

Πρωτού δημιουργηθούν οι παραπάνω 2 Συναρτήσεις, αρχικοποιήσαμε τους εξής μετρητές:

- Θέσαμε όπου: `private String startGame = "X"`; με σκοπό να ξεκινάει το παιχνίδι πρώτος ο Player X.
- Θέσαμε όπου: `private int XCount = 0`; έτσι ώστε το σκορ του PlayerX να είναι αρχικά μηδέν.
- Τέλος, θέσαμε όπου: `private int OCount = 0`; έτσι ώστε το σκορ του PlayerO να είναι και αυτό αρχικά μηδέν.

```
private JFrame frame;  
  
private void btn1ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(btn1.getText().isEmpty()) {  
        btn1.setText(startGame);  
        if(btn1.getText() == "X") {  
            btn1.setForeground(Color.GREEN);  
        }  
        else  
            btn1.setForeground(Color.MAGENTA);  
        if(startGame.equalsIgnoreCase("X")) {  
            TurnDisplay.setText("Player O");  
            TurnDisplay.setForeground(Color.MAGENTA);  
        }  
        else {  
            TurnDisplay.setText("Player X");  
            TurnDisplay.setForeground(Color.GREEN);  
        }  
        ChooseAPlayer();  
        winningGame();  
    }  
}
```

Εικόνα 10: btn1ActionPerformed().

Έπειτα, δημιουργείτε ένα Γεγονός (Event) για το Button1. Για την αποφυγή Λαθών (Errors/Bugs), χρησιμοποιήσαμε μία εξωτερική συνθήκη μέσω της δομής ελέγχου `if`, προκειμένου το γεγονός αυτό να συμβαίνει μόνο εάν το Text του Button1 είναι άδειο. Εάν είναι άδειο τότε:

- Το Text αυτό παίρνει την τιμή της μεταβλητής `startGame`.
- Μέσω μίας επιπλέον δομής `if`:
 - Εάν το Text του Button1 είναι "X" τότε: Το χρώμα της γραμματοσειράς του Button1 αλλάζει σε πράσινο (GREEN).
 - Διαφορετικά: Το χρώμα της γραμματοσειράς γίνεται μωβ (MAGENTA).

- Επίσης, εξετάσαμε το περιεχόμενο του startGame, όπου:
 - Εάν είναι "X" τότε: Το κείμενο του TurnDisplay γίνεται "Player O" ΚΑΙ το χρώμα της γραμματοσειράς αλλάζει σε μωβ (MAGENTA).
 - Διαφορετικά: Το κείμενο του TurnDisplay γίνεται "Player X" ΚΑΙ το χρώμα της γραμματοσειράς αλλάζει σε πράσινο (GREEN).

*Τα παραπάνω θα ισχύουν για κάθε ένα από τα 9 JButtons της εφαρμογής.
Επομένως, με αντίστοιχο τρόπο, προγραμματίσαμε άλλες 8 Συναρτήσεις (Events)
εφαρμόζοντας την ίδια διαδικασία όπως αυτή για το Γεγονός (Event) του Button1.

```

public int winningflag = 0;
public int BCount = 0;

private void winningGame() {
    String Button1 = btn1.getText();
    String Button2 = btn2.getText();
    String Button3 = btn3.getText();
    String Button4 = btn4.getText();
    String Button5 = btn5.getText();
    String Button6 = btn6.getText();
    String Button7 = btn7.getText();
    String Button8 = btn8.getText();
    String Button9 = btn9.getText();

    BCount++;

    //X Tic Tac Toe Conditions.
    //ΟΡΙΖΟΝΤΙΑ: <---
    if(Button1=="X" && Button2=="X" && Button3=="X") {
        JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
        XCount++;
        gameScore();
        winningflag = 1;
        BCount = 0;
    }
    if(Button4=="X" && Button5=="X" && Button6=="X") {
        JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
        XCount++;
        gameScore();
        winningflag = 1;
        BCount = 0;
    }
    if(Button7=="X" && Button8=="X" && Button9=="X") {
        JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
        XCount++;
        gameScore();
        winningflag = 1;
        BCount = 0;
    }
    //ΚΑΤΑΚΟΡΥΦΑ: <---
    if(Button1=="X" && Button4=="X" && Button7=="X") {
        JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
        XCount++;
        gameScore();
        winningflag = 1;
        BCount = 0;
    }
    if(Button2=="X" && Button5=="X" && Button8=="X") {
        JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
        XCount++;
        gameScore();
        winningflag = 1;
        BCount = 0;
    }
    if(Button3=="X" && Button6=="X" && Button9=="X") {
        JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
        XCount++;
    }
}

```

Εικόνα 11: winningGame() - Part1.

Τελευταίο και σημαντικότερο βήμα είναι ο προγραμματισμός των περιπτώσεων (οριζόντιων, κατακόρυφων και πλάγιων) όπου ένας παίκτης του παιχνιδιού κάνει Τρίλιζα. Επίσης σημαντική είναι και η εμφάνιση του νικητή (εάν υπάρξει). Για την πραγματοποίηση των παραπάνω, λοιπόν, δημιουργήσαμε μία Συνάρτηση με όνομα **winningGame()**, *Εικόνα 11*, προγραμματίζοντάς την ως εξής:

- Αρχικά, εκτός της Συνάρτησης, δηλώσαμε 2 public και τύπου integer (int) μεταβλητές (winningflag και BCount) και τις θέσαμε αρχικά 0.

Ξανασημειώνεται πως το BCount είναι για το σύνολο των Buttons που έχουν πατηθεί για την περίπτωση ισοπαλίας, ενώ το winningflag για να σημειώνει νίκη έτσι ώστε παρακάτω να μας βοηθήσει να μηδενίσουμε τα Text των Buttons.

- Εντός της Συνάρτησης, δηλώσαμε 9 τύπου String μεταβλητές (Button1, Button2, Button3, ... κλπ.), οι οποίες περιέχουν το Text κάθε Button. Π.χ String Button1 = btn1.getText(); String Button2 = btn2.getText(); κλπ.
- Στην συνέχεια, αρχίσαμε τον προγραμματισμό των περιπτώσεων σχηματισμού Τρίλιζας για το "X". Έτσι, με μία απλή δομή ελέγχου if, σχηματίσαμε την 1^η οριζόντια περίπτωση ως εξής:

Εάν το Button1 είναι ("X") ΚΑΙ το Button2 είναι ("X") ΚΑΙ το Button3 είναι ("X") τότε:

- ☐ Εμφανίσαμε ενημερωτικό μήνυμα ότι νίκησε ο Player X.
- ☐ Αυξήσαμε τον μετρητή XCount κατά 1: XCount++;
- ☐ Καλέσαμε την Συνάρτηση gameScore(); για εγγραφή των σκόρ.
- ☐ Θέσαμε όπου: winningflag = 1; για σημείωση νίκης.
- ☐ Θέσαμε όπου: BCount = 0; για μηδενισμό του μετρητή ισοπαλίας.

*Ομοίως θα προγραμματίσουμε και όλες τις άλλες κάθετες, οριζόντιες και πλάγιες περιπτώσεις σχηματισμού Τρίλιζας, τόσο για το "X" όσο και για το "O". Η συνέχεια των περιπτώσεων αυτών φαίνεται παραπάνω στην *Εικόνα 11* και παρακάτω στις *Εικόνα 12* και *Εικόνα 13*.

```

//ΠΑΡΤΙΑ: <<---
if(Button1=="X" && Button5=="X" && Button9=="X") {
    JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    XCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}
if(Button3=="X" && Button5=="X" && Button7=="X") {
    JOptionPane.showMessageDialog(this, "Player X Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    XCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}

////////////////////////////////////

//"O" Tic Tac Toe Conditions.
//ΟΡΙΖΟΝΤΙΑ: <<---
if(Button1=="O" && Button2=="O" && Button3=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}
if(Button4=="O" && Button5=="O" && Button6=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}
if(Button7=="O" && Button8=="O" && Button9=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}

//ΚΑΤΑΚΟΡΥΦΑ: <<---
if(Button1=="O" && Button4=="O" && Button7=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}
if(Button2=="O" && Button5=="O" && Button8=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}
if(Button3=="O" && Button6=="O" && Button9=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
}

```

Εικόνα 12: winningGame() – Part2.

```

}
if(Button3=="O") && Button6=="O" && Button9=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}
//ΠΑΡΤΙΑ: <---
if(Button1=="O") && Button5=="O" && Button9=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}
if(Button3=="O") && Button5=="O" && Button7=="O") {
    JOptionPane.showMessageDialog(this, "Player O Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    OCount++;
    gameScore();
    winningflag = 1;
    BCount = 0;
}

if(winningflag == 1){
    btn1.setText("");
    btn2.setText("");
    btn3.setText("");
    btn4.setText("");
    btn5.setText("");
    btn6.setText("");
    btn7.setText("");
    btn8.setText("");
    btn9.setText("");

    winningflag = 0;
}

if(BCount==9){
    JOptionPane.showMessageDialog(this, "Nobody Wins", "Tic Tac Toe", JOptionPane.INFORMATION_MESSAGE);
    btn1.setText("");
    btn2.setText("");
    btn3.setText("");
    btn4.setText("");
    btn5.setText("");
    btn6.setText("");
    btn7.setText("");
    btn8.setText("");
    btn9.setText("");
    BCount = 0;
}
}

```

Εικόνα 13: winningGame() – Part3.

Τέλος, μετά την πραγματοποίηση των περιπτώσεων σχηματισμού Τρίλιζας, χρησιμοποιήσαμε 2 δομές if. Η 1^η if περιλαμβάνει την μεταβλητή winningflag, στη συνθήκη, και έχει ως στόχο τον μηδενισμό του Text κάθε Button μετά τη νίκη κάποιου παίκτη. Η 2^η if περιλαμβάνει το BCount, η οποία, εκτός από τον μηδενισμό του Text κάθε Button μετά τη νίκη κάποιου παίκτη, έχει και ως στόχο την εμφάνιση ενημερωτικού μηνύματος ισοπαλίας. Η πραγματοποίηση αυτών φαίνεται στην *Εικόνα 13*.