



# Quiz The Game

01.11.2020-30.11.2020

Ξενοφών Νικόλαος Πάντσος

Πολυχρόνης Βαρβάρης

Τυριτίδης Γιώργος

## Η λειτουργία της ομάδας

Η ομάδα αποτελείται από 3 φοιτητές του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙΠΑΕ. Στην αρχή, έγινε η γνωριμία των μελών της ομάδας καθώς όλα τα άτομα ήταν άγνωστα μεταξύ τους με διαφορετικές γνώσεις και προσδοκίες. Η επικοινωνία μεταξύ των μελών γινόταν στον Server της ομάδας στο [Discord](#) με συχνές βιντεοκλήσεις. Το κάθε άτομο ξεχωριστά αναλάμβανε διάφορα μέρη του project, φυσικά με την βοήθεια όλων των μελών όπου χρειαζόταν. Τα μέλη μετά την ολοκλήρωση του κάθε "task" κοινοποιούσαν την εξέλιξη της εργασίας στο "[Google Drive](#)". Όλα τα "task" ανατίθονταν στο "[trello](#)". Επιλέχθηκε αυτός ο τρόπος εργασίας έτσι ώστε να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα στον χρόνο που διαθέταμε.

## Λίγα λόγια για το project

Στα πλαίσια της εργασίας του μαθήματος Αλληλεπίδραση Ανθρώπου Μηχανής επιλέχθηκε η δημιουργία ενός Quiz. Το Quiz αποτελείται από 10 κατηγορίες ερωτήσεων όπου ο χρήστης καλείται να επιλέξει τις 2 από αυτές και να απαντήσει σε 10 ερωτήσεις που αντιπροσωπεύουν τις κατηγορίες. Στο τέλος του παιχνιδιού γίνεται ορατό το σκορ που επετεύχθει από τον χρήστη.

## Στόχοι

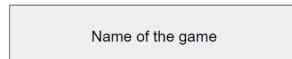
1. Να γίνει ένα ψυχαγωγικό παιχνίδι ερωτήσεων για όλες τις ηλικίες.
2. Η εφαρμογή να διαθέτει μια φιλική διεπαφή με τον χρήστη.
3. Να υπάρχει μια μεγάλη γκάμα κατηγοριών και ερωτήσεων.

## Εργαλεία

- I. Java Swing
- II. Netbeans
- III. Google Drive
- IV. Discord
- V. Trello
- VI. Google Slides
- VII. Google Docs
- VIII. Adobe Photoshop 2020

## Ανάλυση πρωτότυπου

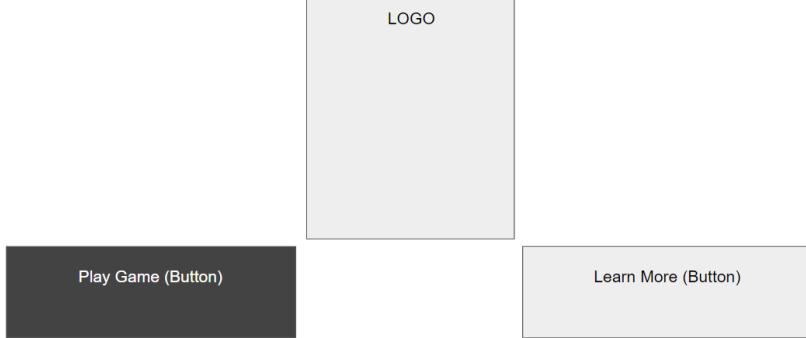
Αφού έγινε η επιλογή της εφαρμογής που θα δημιουργηθεί, στην συνέχεια συζητήθηκαν και αναλύθηκαν οι ιδέες του καθενός για τις προδιαγραφές, για την διεπαφή και για τις μεθόδους προγραμματισμού που θα χρησιμοποιηθούν. Έπειτα από μελέτη της κάθε ιδέας επιλέχθηκαν οι πιο ορθολογικές. Σύμφωνα με αυτές τις ιδέες κατασκευάστηκε το πρωτότυπο. Το πρωτότυπο είναι της μορφής "Story Board". Δημιουργήθηκε στο "[Google Slides](#)" όπου η κάθε διαφάνεια αντιπροσωπεύει μια εικόνα που παραπέμπουν στην Αλληλεπίδραση της εφαρμογής με τον χρήστη.



*Eikόνα 1.1 : Start page*



*Eikόνα 1.2: Start page*



Στην Εικόνα 1.1 διακρίνεται η μορφή της διεπαφής με τα 2 button Play Game, More Info και ο τίτλος. Ενώ στην Εικόνα 1.2 παρουσιάζεται η αλληλεπίδραση του χρήστη και της εφαρμογής με το πάτημα του κουμπιού Play Game όπου σκιαγραφείται. Στην Εικόνα 3 φαίνεται η διάδραση της διεπαφής με το πάτημα του κουμπιού. Αυτη είναι η λογική ώστε να μπορεί κάποιος να διαβάσει το πρωτότυπο. Με το πάτημα του κουμπιού Play Game, ο

χρήστης οδηγείται στις κατηγορίες, όπου πρέπει να επιλεγούν 2 από τις 10. Αφού πραγματοποιηθεί η επιλογή 2 κατηγοριών και πατηθεί το κουμπί next παρουσιάζεται το κύριο μέρος του παιχνιδιού.

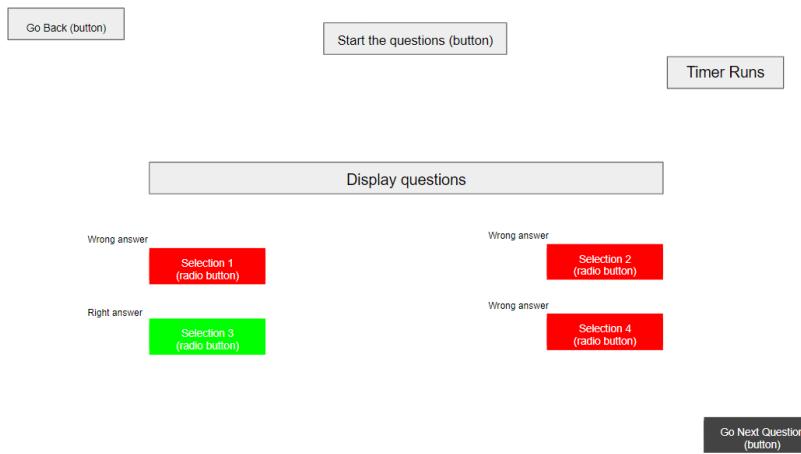
The screenshot shows a user interface for selecting categories. At the top left is a 'Go Back (button)' and at the top center is a 'Choose 2 categories (info)' button. Below these are ten categories arranged in two rows of five. The categories are labeled: Category 1 (checkbox), Category 6 (checkbox) in the first row; Category 2 (checkbox), Category 7 (checkbox) in the second row; Category 3 (checkbox), Category 8 (checkbox) in the third row; Category 4 (checkbox), Category 9 (checkbox) in the fourth row; and Category 5 (checkbox), Category 10 (checkbox) in the fifth row. A large 'Next (button)' is located at the bottom right.

*Eικόνα 2: Categories page*

The screenshot shows a user interface for a quiz. At the top left is a 'Go Back (button)', in the center is a 'Start the questions (button)', and at the top right is a 'Timer Starts' button. Below these is a 'Display questions' button. Underneath are four selection options: 'Selection 1 (radio button)', 'Selection 2 (radio button)', 'Selection 3 (radio button)', and 'Selection 4 (radio button)'. At the bottom right is a 'Go Next Question (button)'.

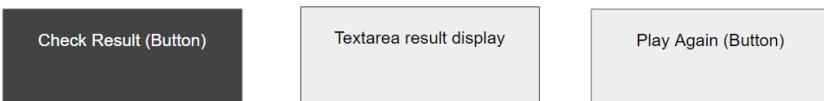
*Eικόνα 3: Quiz page*

Ο χρήστης καλείται πλέον να πατήσει το κουμπί start, ετσι ώστε να ξεκινήσει το παιχνίδι. Έπειτα εμφανίζονται οι ερωτήσεις και οι απαντήσεις. Στην συνέχεια επιλέγει μια απαντηση, η εφαρμογή εμφανίζει ποιά είναι η σωστή και με το κουμπί next οδηγείται στην επόμενη ερώτηση. Αυτό, επαναλαμβάνεται για 10 ερωτήσεις.



*Eikόνα 4: Wrong-Right answers*

Αφού απαντηθούν και οι 10 ερωτήσεις ή έχει λήξει ο χρόνος των 3 λεπτών του παιχνιδιού, ο χρήστης καλείται να απαντήσει το κουμπί result, ώστε να δει το τελικό σκορ του. Το τελικό σκορ εμφανίζεται σε ένα textbox, αφού πατηθεί το κουμπί check score, με κλίμακα από το 0 έως το 10, ανάλογα το πόσες ερωτήσεις έχει απαντήσει σωστά.



*Eikόνα 5: Result page*

## Ανάλυση Εφαρμογής

Μετά την ολοκλήρωση του πρωτούπου ξεκίνησε η ανάπτυξη της εφαρμογής στο NetBeans. Μετά το κάθε meeting, το κάθε μέλος της ομάδας ξεχωριστά αναλάμβανε ένα task. Αφού ολοκληρώνονταν όλα τα task πραγματοποιούταν ξανά ένα meeting ώστε να φανεί η πρόοδος του καθενός, να κατανοηθεί ο κώδικάς του και για ενωθούν όλα τα task. Χρησιμοποιήθηκε η μέθοδος του οπτικού προγραμματισμού και drag and drop.

### I. Start Page



*Εικόνα 6: Start page finalized*

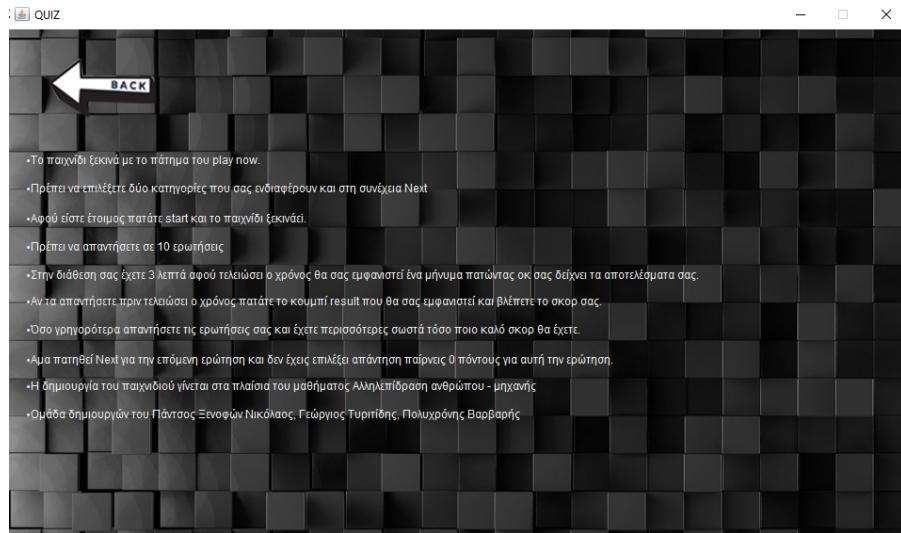
Στο Start Page χρησιμοποιήθηκαν 2 Action Events, το playButtonActionPerformed και το infoButtonActionPerformed. Η μέθοδος playButtonActionPerformed χρησιμοποιείται για το πάτημα του κουμπιού Play Now , όπου σε μεταφέρει στο Categories jFrame. Αντίστοιχα, η infoButtonActionPerformed χρησιμοποιείται για το πάτημα του κουμπιού More Info, όπου σε μεταφέρει στο Info jFrame.

```
private void playButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new categories().setVisible(true);
}

private void infoButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new info().setVisible(true);
}
```

*Εικόνα 7: Play-Info Buttons Actions*

## II. More Info



*Eikόνα 8: More Info page finalized*

Στο συγκεκριμένο JFrame αναγράφονται πληροφορίες και κανόνες σχετικά με το παιχνίδι. Για τις πληροφορίες χρησιμοποιήθηκαν 10 Label ιδίου τύπου με αρίθμηση κουκίδων. Επίσης, χρησιμοποιείται το κουμπί Back Button όπου σε οδηγεί πίσω στο Start Page.

```
private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new StartPage().setVisible(true);
}
```

*Eikόνα 9: Μέθοδος backButton*

### III. Categories



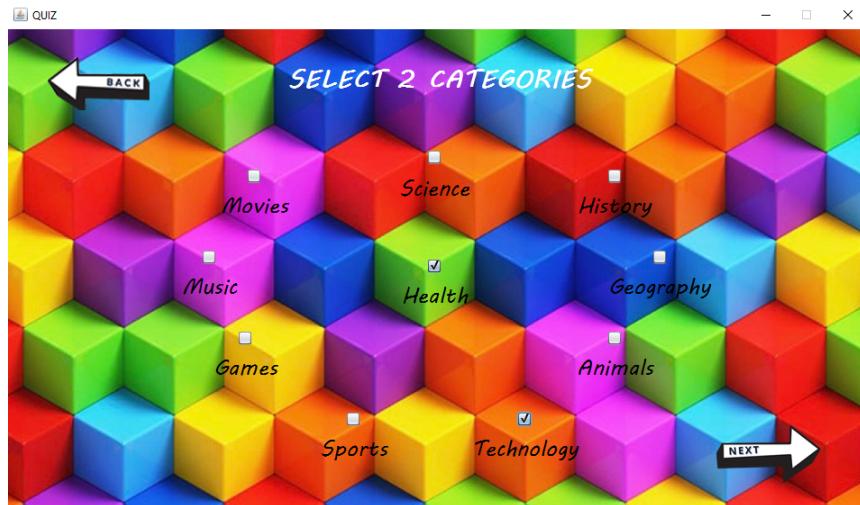
Εικόνα 10: CheckBox Categories

Στο συγκεκριμένο JFrame εμφανίζονται οι κατηγορίες του Quiz σε μορφή Check Box, όπου ο χρήστης καλείται να επιλέξει τις 2 από τις 10. Αυτό επιτυγχάνεται με την παρακάτω μέθοδο για κάθε Check Box αντίστοιχα.

```
private void healthCheckBoxActionPerformed(java.awt.event.ActionEvent evt) {
    if (healthCheckBox.isSelected()){
        if (index1 == 0){
            index1=8;
        }
        else if(index2 == 0){
            index2=8;
        }
        flag++;
    }
    else{
        if (index1==8){
            index1=0;
        }
        else if (index2==8){
            index2=0;
        }
        flag--;
    }
    if (flag==2){
        nextButton.setVisible(true);
    }
    else if (flag<2 || flag>2){
        nextButton.setVisible(false);
    }
}
```

Εικόνα 11: Μεθοδος CheckBox Categories

Για κάθε Check Box (κατηγορία) αντιστοιχεί ένας αριθμός από το 1 μέχρι το 10. Οι μεταβλητές που έχουμε ορίσει είναι οι ακέραιες index1, index2 όπου οι τιμές που παίρνουν είναι ανάλογες με το Check Box που έχει επιλεχθεί και η flag η οποία δέχεται μόνο ακέραιες τιμές και λειτουργεί σαν ενας μετρητής που μετράει πόσα Check Boxes είναι επιλεγμένα ώστε να κάνει ορατό το κουμπί Next.



Εικόνα 12: Choose 2 Categories

To Back Button σε αυτό το Frame μας παραπέμπει στο Start Page.

```
private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new StartPage().setVisible(true);
}
```

Εικόνα 13: Μέθοδος backButton

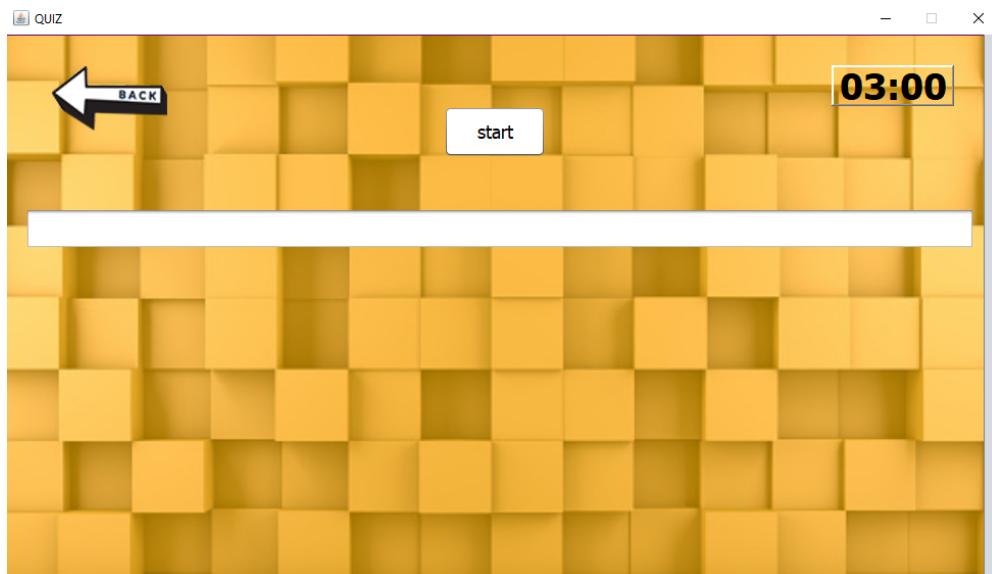
Τέλος, το Next Button εμφανίζεται αφού επιλεχθούν 2 μόνο κατηγορίες και με το πάτημά του παίρνουν τις ανάλογες τιμές οι 2 παράμετροι που αντιστοιχούν στην κάθε κατηγορία, όπου θα χρησιμοποιηθούν στο Quiz jFrame.

```
private void nextButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new Quiz(index1, index2).setVisible(true);
}
```

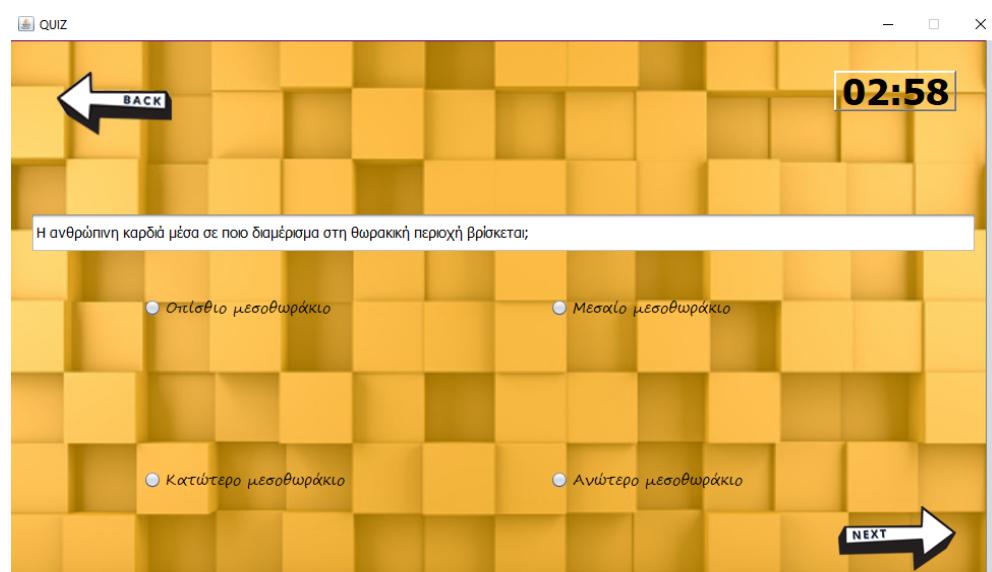
Εικόνα 14: Μέθοδος nextButton

## IV. Quiz

Το jFrame που ονομάζεται Quiz, αποτελεί και το βασικό μέρος της εφαρμογής καθώς από εκεί παίζεται το παιχνίδι. Μέσα στο συγκεκριμένο Frame βρίσκονται οι ερωτήσεις και οι απαντήσεις των ερωτήσεων, καθώς και ένα timer το οποίο μετράει αντίστροφα τον χρόνο του παίκτη. Με το πάτημα του κουμπιού start το παιχνίδι ξεκινάει και εμφανίζονται 5 τυχαίες ερωτήσεις ανάλογα με τις κατηγορίες που επιλέχθηκαν.



Εικόνα 15.1: Start Quiz



Εικόνα 15.2: Quiz Choices

Αφού επιλεχθεί η απάντηση που θεωρείται σωστή, εμφανίζεται ποια ήταν η σωστή με πράσινο χρώμα, ενώ όλες οι υπόλοιπες με κόκκινο. Έπειτα το κουμπί next κατευθύνει τον χρήστη στην επόμενη ερώτηση. Αυτό επαναλαμβάνεται μέχρι να απαντηθούν οι 10 ερωτήσεις ή μέχρι να τελειώσει ο χρόνος που έχει στην διάθεσή του ο παίκτης, τότε εμφανίζεται το κουμπί result στην θέση του κουμπιού next. Το κουμπί result κατευθύνει τον χρήστη στο Result JFrame.

Οι ερωτήσεις αποθηκεύτηκαν σε έναν μονοδιάστατο πίνακα, όπου κάθε 10 κελιά αντιπροσωπεύουν και την ανάλογη κατηγορία των ερωτήσεων.

```
String[] questions = {
    //Sports
    "Σε πόσους τελικούς Μουντιάλ έχει αγωνιστεί ο Βραζιλιάνος Καφού;",  

    "Ποιος είναι ο γηραιότερος παγκόσμιος πρωταθλητής στην ιστορία του μποξ?",  

    "Ποιο είναι το πραγκόσμιο ρεκόρ που κατέχει ο Γιουσέιν Μπολτ στα 100 μέτρα?",  

    "Σε ποιά ομάδα έχει παίξει ο Michael Jordan εκτός από τους Chicago Bulls?",  

    "Πόσα σερί πρωταθλήματα κατέκτησε με τη Φεράρι ο Michael Schumacher στη Φόρμουλα 1?",  

    "Πόσα λεπτά κρατάει θεωρικά μία περίοδος στην υδατοσφαίριση?",  

    "Κάθε πόσα χρόνια γίνεται το Ευρωπαϊκό Πρωτάθλημα ποδοσφαίρου?",  

    "Ποιό από τα παρακάτω αθλήματα δεν ανοίκει στο Δέκαθλο?",  

    "Σε ποιά χώρα εφευρέθηκε το Κρίκετ?",  

    "Ποιό από τα παρακάτω δεν θεωρείται αθληματοδότης?",  

    // Zoo
    "Ποιανούς ζώους η εγκυμοσύνη κρατάει 5 μήνες?",  

    "Με πόσα θηλυκά ζευγαρώνει ένας κύκνος?",  

    "Κάθε πότε μπορεί να γεννήσει μια κουνέλαια?",  

    "Πόσα μωρά γεννώνει μια γουρούνα σε κάθε γέννα?",  

    "Πόσα χρόνια ζει μια χελώνα της έπρασ?",  

    "Πόσα πόδια έχει μια ακρίδα?",  

    "Πόσα περίπου μύδια τρώει ένας θαλάσσιος ιππος κάθε μέρα?",  

    "Ποιο ζώο πηδά έφα από το νερό για να επικοινωνήσει με άλλους του είδους του?",  

    "Ποιο πουλί μπορεί να πετάξει ευθεία, κάτω και προς τα πίσω?",  

    "Ποιο ζώο είναι το μόνο ιπτάμενο θηλαστικό?",
```

*Eikόνα 16: Questions Array*

Οι επιλογές των απαντήσεων τοποθετήθηκαν σε έναν τρισδιάστατο πίνακα με τον ίδιο ακριβώς τρόπο.

```
String[][] options = {
    //Sport
    {"Κονένων", "2", "3", "4"},  

    {"Muhammad Ali", "Bernard Hopkins", "George Foreman", "Shane Mosley"},  

    {"9.68", "9.63", "8.90", "9.58"},  

    {"Washington Wizards", "Denver Nuggets", "Brooklyn Nets", "Golden State Warriors"},  

    {"7", "3", "5", "0"},  

    {"8", "10", "15", "4"},  

    {"2", "1", "6"},  

    {"Δισκοβολία", "Άλμα επι κοντό", "Άρση βαρών", "Σφαιροβολία"},  

    {"Γολλία", "Ισαλνδία", "Τερμανία", "Αγγία"},  

    {"Εκάκι", "Βελάκια", "Μπόουλινγκ", "Πόκερ"},  

    //Zoo
    {"Του αλόγου", "Της κατσίκας", "Της αγελάδας", "Του γουρουνιού"},  

    {"Με 1", "Με 3", "Με 5", "Με 7"},  

    {"Κάθε μήνα", "Κάθε 2 μήνες", "Κάθε 3 μήνες", "Κάθε 4 μήνες"},  

    {"1-3", "4-6", "6-9", "9-12"},  

    {"1-20", "20-50", "50-100", "Πάνω από 100"},  

    {"5", "6", "2", "4"},  

    {"5,000", "10,000", "12,000", "16,000"},  

    {"Ιπτάμενο Ψάρι", "Κροκόδειλος", "Φάλαινα", "Καρχαρίας"},  

    {"Περιστέρι", "Κολίβριο πουλί", "Αετός", "Παπαγάλος"},  

    {"Νυχτερίδα", "Ιπτάμενος σκίουρος", "Πελαργός", "Αετός"},
```

*Eikόνα 17: Options Array*



Η σωστή επιλογή των απαντήσεων αποθηκεύτηκαν σε έναν μονοδιάστατο πίνακα με όνομα answers. Η μορφή τους είναι A,B,C και D. Όλοι οι πίνακες μεταξύ τους είναι παράλληλοι, π.χ. η ερώτηση που βρίσκεται στο 8ο κελί του πίνακα questions αντιστοιχεί με τις επιλογές της 8ης γραμμής του πίνακα options και αντίστοιχα με το 8ο κελί του answers.

```
char[] answers = {
    //Sports
    'C',
    'B',
    'D',
    'A',
    'C',
    'A',
    'A',
    'C',
    'D',
    'D',
    //Zoo
    'B',
    'A',
    'A',
    'D',
    'D',
    'B',
    'A',
    'C',
    'B',
    'A',
```

*Eikόνα 17: Answers Array*

Έχει οριστεί η μεταβλητή flag όπου με το πάτημα των κουμπιών start και next αυξάνεται κατά μία μονάδα μέχρι να γίνει 10 και τότε εμφανίζεται το result button.

Η catemethod είναι μια μέθοδος που χρησιμοποιεί τις παραμέτρους από το JFrame των categories, ώστε να ορίσει την αρχή και το τέλος των ερωτήσεων σε κάθε κατηγορία. Έτσι, βρίσκουμε 4 μεταβλητές την min1 όπου χαρακτηρίζει την πρώτη ερώτηση της κατηγορίας και την max1 που χαρακτηρίζει την τελευταία ερώτηση. Αντίστοιχα, το min2 και max2 για την δεύτερη κατηγορία που επιλέχθηκε.

```

private void catemethod() {
    if(cat1==1) {
        min1=0;
        max1=9;}
    else if (cat1==2) {
        min1=10;
        max1=19;}
    else if (cat1==3) {
        min1=20;
        max1=29;}
    else if (cat1==4) {
        min1=30;
        max1=39;}
    else if (cat1==5) {
        min1=40;
        max1=49;}
    else if (cat1==6) {
        min1=50;
        max1=59;}
    else if (cat1==7) {
        min1=60;
        max1=69;}
    else if (cat1==8) {
        min1=70;
        max1=79;}
    else if (cat1==9) {
        min1=80;
        max1=89;}
    else if (cat1==10) {
        min1=90;
        max1=99;}}
    if(cat2==1) {
        min2=0;
        max2=9;}
    else if (cat2==2) {
        min2=10;
        max2=19;}
    else if (cat2==3) {
        min2=20;
        max2=29;}
    else if (cat2==4) {
        min2=30;
        max2=39;}
    else if (cat2==5) {
        min2=40;
        max2=49;}
    else if (cat2==6) {
        min2=50;
        max2=59;}
    else if (cat2==7) {
        min2=60;
        max2=69;}
    else if (cat2==8) {
        min2=70;
        max2=79;}
    else if (cat2==9) {
        min2=80;
        max2=89;}
    else if (cat2==10) {
        min2=90;
        max2=99;}}
}

```

Εικόνα 18: Μέθοδος catamethod

Η randommethod1 χρησιμοποιείται για τη δημιουργία μιας τυχαίας μεταβλητής. Η τυχαία μεταβλητή βρίσκεται ανάμεσα στα όρια τα οποία έχουν τεθεί στην catemethod (min1,min2,max1,max2). Όταν το flag είναι μικρότερο του 6, οι τυχαίες μεταβλητές είναι ανάμεσα στα όρια του min1 και max1, δηλαδή την πρώτη κατηγορία ερωτήσεων, ενώ όταν είναι μεγαλύτερο του 5, οι τιμές είναι ανάμεσα στα όρια του min2 και max2 δηλαδή στην δεύτερη κατηγορία ερωτήσεων. Οι τυχαίες μεταβλητές δημιουργούνται μέσα σε μια do while, ώστε να αποφευχθούν οι διπλότυπες μεταβλητές.

```

private int randommethod1(){
    int index1;
    Random rand = new Random();
    if(flag<6){

        do{
            index1=rand.nextInt((max1-min1)+1)+min1;
        }while(pinl[0]==index1 || pinl[1]==index1 || pinl[2]==index1 || pinl[3]==index1 || pinl[4]==index1);
        pinl[i]=index1;
        i++;
    }
    return index1;
    }else
    {

        do{
            index1=rand.nextInt((max2-min2)+1)+min2;
        }while(pinl[5]==index1 || pinl[6]==index1 || pinl[7]==index1 || pinl[8]==index1 || pinl[9]==index1);
        pinl[i]=index1;
        i++;
    }
    return index1;
}
}

```

*Εικόνα 19: Μέθοδος randommethod1*

Στην εφαρμογή, χρησιμοποιείται ένα χρονόμετρο με αντίστροφη μέτρηση ώστε όταν μηδενιστεί να εμφανίζεται μια ειδοποίηση στον πάικτη ότι τελείωσε ο χρόνος που έχει στην διάθεση του για να απαντήσει στις ερωτήσεις. Η μέθοδος που χρησιμοποιείται για το χρονόμετρο είναι η Timer T με την βοήθεια των βιβλιοθηκών javax.swing.Timer και java.awt.event.\*. Το πρώτο βήμα είναι να αρχικοποιηθούν οι τιμές των μεταβλητών min (λεπτά) και sec (δευτερόλεπτα), αλλά πρέπει να φτιαχτεί και μια τοπική boolean μεταβλητή με όνομα check, που θα μας βοηθήσει στου ελέγχους. (εικόνα 20)

```

///////////
//Setting Timer
int min=3;
int sec=0;
Timer T= new Timer(1000, new ActionListener(){

    public void actionPerformed(ActionEvent e){
        boolean check=true;//A flag that helps us with if statements
        if (sec==0){ //Checking the seconds and passing minutes
            min--;
            sec=60;

        }
        if (min<0){//If we reach this point that means we run out of time
            JOptionPane.showMessageDialog(rootPane, "Times Up", "Game Over", 0); //A popup Error when time is over
            min=sec=0; //Setting Minutes and seconds back to 0 and stopping the timer
            T.stop();
            new result(correct_guesses).setVisible(true); //Going to result page
            setVisible(false);
        }else{
            sec--;
            if (sec<10){//This statement is for printing 05:09 etc and not 05:9 etc
                jTimer.setText("0"+min+":"+ "0"+sec);
                check=false;
            }
            if (check){ //We are all set if we reach this point
                jTimer.setText("0"+min+":"+sec);
            }
        }
    }
});

```

*Εικόνα 20: Timer*

Μόλις προβληθεί το Quiz jFrame, ο χρήστης καλείται να πατήσει το κουμπί start για να ξεκινήσει το παιχνίδι και η αντίστροφη μέτρηση. Αφού πατηθεί το κουμπί, το flag παίρνει την τιμή 1 έπειτα καλείται η catemethod (εικόνα 18), η μεταβλητή index παίρνει την τιμή που επιστρέφει η μέθοδος randommethod1 (εικόνα 19) και στην συνέχεια δίνονται οι ανάλογες τιμές στο questionText και στα radio buttons που αντιπροσωπεύουν τις επιλογές. Έπειτα το Start Button, γίνεται μη ορατό και εμφανίζεται το next button (εικόνα 21).

```
private void startButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    T.start();  
    flag=1;  
    catemethod();  
    index=randommethod1();  
    questionText.setText("Question "+(index));  
    questionText.setText(questions[index]);  
    aButton.setText(options[index][0]);  
    bButton.setText(options[index][1]);  
    cButton.setText(options[index][2]);  
    dButton.setText(options[index][3]);  
    startButton.setVisible(false);  
    startButton.setEnabled(false);  
    nextButton.setVisible(true);  
    nextButton.setEnabled(true);  
    aButton.setVisible(true);  
    bButton.setVisible(true);  
    cButton.setVisible(true);  
    dButton.setVisible(true);  
}
```

Εικόνα 21: Μέθοδος startButtonsActionsPerformed

Για κάθε radio button των επιλογών υπάρχει μια μέθοδος. Η κάθε μέθοδος ελέγχει αν η απάντηση του χρήστη είναι σωστή σύμφωνα με τον πίνακα answers (εικόνα 17). Στην συνέχεια, αφού γίνει η επιλογή φαίνονται με κόκκινο χρώμα οι λανθασμένες επιλογές, ενώ η σωστή επιλογή τονίζεται με πράσινο. Τέλος, στην συγκεκριμένη μέθοδο, εφόσον η απάντηση που δόθηκε είναι η σωστή αυξάνεται η μεταβλητή correct\_guesses κατά μια μονάδα και έτσι καταγράφονται οι σωστές απαντήσεις του χρήστη (εικόνα 22).

```
private void aButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    aButton.setEnabled(false);  
    bButton.setEnabled(false);  
    cButton.setEnabled(false);  
    dButton.setEnabled(false);  
  
    if(evt.getSource() == aButton)  
    {  
        answer= 'A';  
        if(answer == answers[index])  
        {  
            correct_guesses++;  
            aButton.setForeground(Color.green);  
            bButton.setForeground(Color.red);  
            cButton.setForeground(Color.red);  
            dButton.setForeground(Color.red);  
        }  
        answer= 'B';  
        if(answer == answers[index])  
        {  
            dButton.setForeground(Color.RED);  
            aButton.setForeground(Color.RED);  
            bButton.setForeground(Color.GREEN);  
            cButton.setForeground(Color.red);  
        }  
        answer= 'C';  
        if(answer == answers[index])  
        {  
            dButton.setForeground(Color.RED);  
            aButton.setForeground(Color.RED);  
            bButton.setForeground(Color.red);  
            cButton.setForeground(Color.GREEN);  
        }  
        answer= 'D';  
        if(answer == answers[index])  
        {  
            dButton.setForeground(Color.green);  
            aButton.setForeground(Color.RED);  
            bButton.setForeground(Color.red);  
            cButton.setForeground(Color.red);  
        }  
    }  
}
```

Εικόνα 22: Right/Wrong Answers



Μετά το πάτημα του start button γίνεται ορατό το κουμπί next. Ο χρήστης, μπορεί να το πατήσει αφού επιλέξει την απάντηση που θεωρεί σωστή ή χωρίς να επιλέξει απάντηση. Τότε, καλείται η μέθοδος nextButtonActionPerformed. Η συγκεκριμένη μέθοδος ελέγχει εάν το flag ίσο με 9 που αυτό σημαίνει ότι έχουν εμφανιστεί όλες οι ερωτήσεις και τότε θα εμφανιστεί το κουμπί result. Εφόσον το flag είναι μικρότερο του 9, η μέθοδος τοποθετεί τις κατάλληλες τιμές στο questionText και στα radio Buttons. Επίσης, επαναφέρει το αρχικό χρώμα των επιλογών και αυξάνει το flag κατά 1 μονάδα.

```
private void nextButtonActionPerformed(java.awt.event.ActionEvent evt) {
    if (flag==9){
        nextButton.setVisible(false);
        resultButton.setVisible(true);
        min=0;sec=0;
        T.stop();
    }

    if(!startButton.equals(evt)){
        flag++;
        index=randommethod1();
        questionText.setText(questions[index]);
        aButton.setText(options[index][0]);
        bButton.setText(options[index][1]);
        cButton.setText(options[index][2]);
        dButton.setText(options[index][3]);
        aButton.setEnabled(true);
        bButton.setEnabled(true);
        cButton.setEnabled(true);
        dButton.setEnabled(true);
        cButton.setForeground(Color.black);
        aButton.setForeground(Color.black);
        bButton.setForeground(Color.black);
        dButton.setForeground(Color.black);
        aButton.setSelected(false);
        bButton.setSelected(false);
        cButton.setSelected(false);
        dButton.setSelected(false);
    }
}
```

*Εικόνα 23: Μέθοδος nextButtonActionPerformed*

Το κουμπί result χρησιμοποιείται για να μας κατευθύνει στο jFrame Result, δημιουργώντας ένα αντικείμενο με τιμή την παράμετρο correct\_guesses.

```
private void resultButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new result(correct_guesses).setVisible(true);
}
```

*Εικόνα 24: Result Button*



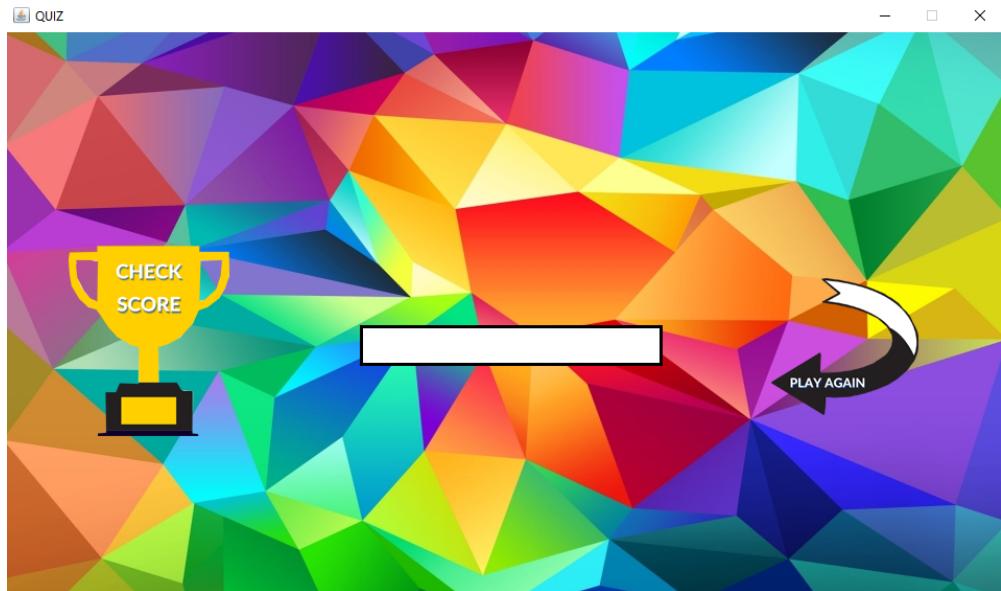
Το κουμπί Back ανακατευθείνει τον χρήστη στο Start Page, ακυρώνοντας το παιχνίδι.

```
private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new StartPage().setVisible(true);
}
```

*Eikόνα 25: Back Button*

## V. Result

Το JFrame, που ονομάζεται Result, χρησιμοποιείται για να δείξει στον χρήστη το τελικό score του. Στο συγκεκριμένο JFrame υπάρχουν τα κουμπιά Check Score και Play Again, όπως και ένα textField που χρησιμοποιείται για την ανάδειξη του score.



*Eikόνα 26: Result page finalized*



Το κουμπί Check Score χρησιμοποιείται για την ανάδειξη του score στο Textfield που πέτυχε ο χρήστης χρησιμοποιώντας την τιμή της παραμέτρου που δόθηκε προηγουμένως.

```
private void checkScoreActionPerformed(java.awt.event.ActionEvent evt) {  
    resultText.setText(" "+ corect+"/10");  
}
```

*Eikόνα 27: Check Score*

Το κουμπί Play Again δίνει την δυνατότητα στον χρήστη να ξαναπαίξει το παιχνίδι επιστρέφοντάς τον στην επιλογή κατηγοριών.

```
private void playagainButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    this.setVisible(false);  
    new categories().setVisible(true);  
}
```

*Eikόνα 28: Play Again Button*

## Εύρεση Ερωτήσεων

Για την εύρεση των 100 ερωτήσεων και των κατάλληλων επιλογών τους, πραγματοποιήθηκε μια απαιτητική έρευνα. Πολλές από τις ερωτήσεις του Quiz έχουν δοθεί από ειδικούς του συγκεκριμένου θέματος. Για παράδειγμα, οι ερωτήσεις που αφορούσαν την κατηγορία Ζωα καταγράφηκαν από κτηνίατρο, ενώ οι ερωτήσεις που αφορούσαν την κατηγορία Υγεία καταγράφηκαν από νοσηλευτή. Οι υπόλοιπες ερωτήσεις βρέθηκαν κυρίως από έρευνα σε ηλεκτρονικές εγκυκλοπαίδειες.

## Κανόνες του παιχνιδιού

Οι κανόνες του παιχνιδιού αναγράφονται στο [More Info](#), ωστόσο θα γραφούν αναλυτικότερα παρακάτω. Ο παίκτης απαγορεύεται να χρησιμοποιήσει την βοήθεια οποιασδήποτε διαδικτυακής μηχανής αναζήτησης για την εύρεση των απαντήσεων ή οποιοδήποτε άλλου τρόπου πέρα από τις προσωπικές τους γνώσεις. Ο χρόνος που έχει στην διάθεση του για να απαντήσει τις 10 ερωτήσεις είναι 3 λεπτά, εφόσον δεν απαντηθούν οι απαντήσεις σε 3 λεπτά δεν θα έχει την δυνατότητα να απαντήσει παραπάνω ερωτήσεις. Ο χρήστης καλείται να επιλέξει μόνο 2 κατηγορίες από τις 10. Τέλος αν πατηθεί το κουμπί next πριν απαντηθεί η ερώτηση τότε η ερώτηση ακυρώνεται.

## Περίληψη

Το project αναπτύχθηκε από μια ομάδα τριών φοιτητών. Αρχικά, έγινε η συζήτηση και η επιλογή της συγκεκριμένης ιδέας και έπειτα κατασκευάστηκε το πρωτότυπο. Αφού έγιναν τα παραπάνω, ξεκίνησε η ανάπτυξη της εφαρμογής από τα μέλη της ομάδας στο NetBeans. Η εφαρμογή είναι παιχνίδι γνώσεων που στοχεύει σε όλες τις ηλικιακές βαθμίδες με 10 διαφορετικές κατηγορίες ερωτήσεων. Τέλος, αποτελείται από 5 JFrame Start Page, More Info, Categories, Quiz και Result.

## Ομάδα

Η συνεργασία μεταξύ των μελών της ομάδας ήταν οργανωμένη και γρήγορη. Η ομάδα είχε τουλάχιστον δύο online συναντήσεις την εβδομάδα, μέσω discord. Σε αυτές τις συναντήσεις ενθαρρύναμε, συζητούσαμε και προτείναμε λύσεις ο ένας στον άλλον. Το Trello χρησιμοποιήθηκε για την σωστή και άμεση επικοινωνία μας. Η πραγματοποίηση του κώδικα ήταν αποτέλεσμα της συνεχόμενης αλληλεπίδρασης που είχαμε σαν μέλη της ομάδας στο drive, στο οποίο ανέβαινε η καινούργια μορφή του προγράμματος.

## Επίλογος

Το παρόν project θα ανέβει στο [GitHub](#) στα προφίλ των φοιτητών έτσι ώστε να μπορεί ο καθένας να πάρει ιδέες από το project μας, να δει μέρος της δουλειάς μας ή και να παίξει το παιχνίδι που δημιουργήσαμε. Για οποιαδήποτε παρατήρηση, διόρθωση ή απορία θα χαρούμε να λάβουμε email σας.

- Πάντσος Ξενοφών Νικόλαος 185251
  - [Email](#), [GitHub](#)
- Βαρβάρης Πολυχρόνης 185152
  - [Email](#), [Github](#)
- Τυριτίδης Γιώργος 185303
  - [Email](#), [Github](#)