

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος **Αντικειμενοστρεφής Ανάπτυξη Εφαρμογών**

<i>Αριθμός εργασίας – Τίτλος εργασίας</i>	<i>Τελική Εργασία</i>
Όνομα φοιτητή	Σοφία Μαρκοπούλου, Μιχαήλ Στυλιανίδης, Κωνσταντίνος Καλογερόπουλος
Αρ. Μητρώου	Π19097, Π19165, Π19057
Ημερομηνία παράδοσης	23/02/2021



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Εισαγωγή	3
2	Περιγραφή του προγράμματος	3
2.1	Παιχνίδι μνήμης	3
2.2	Σκακιέρα	4
2.3	Διαχείριση Covid-19	6
3	Επίδειξη των εφαρμογών	8



1 Εισαγωγή

Οι πρώτες δύο εφαρμογές υλοποιήθηκαν σε windows forms, ενώ η τρίτη σε ASP .NET.

2 Περιγραφή του προγράμματος

2.1 Παιχνίδι μνήμης

Η εφαρμογή ξεκινάει με την FormMain και εμφανίζεται το μενού που οδηγεί σε όλα τα υπόλοιπα τμήματα της εφαρμογής. Το κάθε τμήμα έχει υλοποιηθεί με τη χρήση user control με μοναδική εξαίρεση το μενού της FormMain. Συγκεκριμένα, υπάρχουν τα user controls:

- a) Login: Η περιοχή της σύνδεσης. Σε αυτό το σημείο ο χρήστης δακτυλογραφεί ένα όνομα και προχωράει στο παιχνίδι
- b) Game: Η περιοχή του παιχνιδιού. Για την υλοποίηση του μηχανισμού του παιχνιδιού έχει δημιουργηθεί η τάξη Card, η οποία κληρονομεί από το PictureBox. Συνολικά στο game section υπάρχουν 16 αντικείμενα της Card. Για την φόρτωση του παιχνιδιού καλείται η μέθοδος LoadGame. Αυτή φορτώνει σε μία λίστα όλα τα paths των εικόνων από τον φάκελο που έχει οριστεί στη στατική τάξη GameSettings. Στη συνέχεια επιλέγει 8 από αυτές τυχαία και φτιάχνει ένα αντίγραφο για καθεμία. Στο τέλος η λίστα με τα 16 πλέον paths ανακατεύεται για άλλη μια φορά και αποθηκεύει ένα path σε κάθε Card και ξεκινάει το παιχνίδι. Όταν ο χρήστης πατάει σε μία κάρτα αυτή ανοίγει και αποθηκεύεται ως ανοιχτή. Μόλις πατήσει την επόμενη συγκρίνει τα paths τους και αν είναι ίδια τότε παραμένουν ανοιχτές και οι 2. Διαφορετικά μετά από μία καθυστέρηση 400 milisecond (υλοποιημένης με Thread) κλείνουν. Μόλις ανοίξουν όλες οι κάρτες αποθηκεύεται ο παίκτης με το όνομα, τις προσπάθειες και το χρόνο που έκανε (τάξη Player) και η εφαρμογή μεταβαίνει στο scoreboard. Κάθε στιγμή πριν το τέλος του παιχνιδιού ο χρήστης έχει τη δυνατότητα να πατήσει esc, ώστε να κάνει παύση και να ανοίξει ένα μενού που περιέχει τις επιλογές: συνέχεια, αλλαγή ονόματος και επιστροφή στο μενού. Εάν επιλέξει μία από τις δύο τελευταίες το παιχνίδι επαναφέρεται στην αρχική κατάσταση και χάνεται η πρόοδος του παίκτη. Για την υλοποίηση του μενού αυτού έχουν δημιουργηθεί 2 φόρμες. Η FormMask και η FormGameMenu.
 - a. FormMask: Αυτή η φόρμα χρησιμοποιείται για να σκοτεινιάσει την FormMain. Αυτό επιτυγχάνεται ορίζοντας το background color σε μαύρο και το Opacity σε 80%.
 - b. FormGameMenu: Η φόρμα που περιέχει το μενού



- c) Scoreboard: Σε αυτή την περιοχή φαίνονται οι προσπάθειες των παικτών με αύξουσα σειρά ανάλογα με το χρόνο. Για την απεικόνιση των παικτών έχουν χρησιμοποιηθεί τα user controls PlayerModel και Champion. Σε αυτά φορτώνονται οι πληροφορίες των παικτών και εμφανίζονται στην οθόνη (Στο κέντρο πάνω ο πρωταθλητής και απο κάτω όλοι οι άλλοι παίκτες). Εάν δε χωράνε όλοι οι παίκτες μπορούν να χρησιμοποιηθούν τα βελάκια για τη μετακίνηση των παικτών δεξιά και αριστερά.
- d) Options: Η περιοχή των ρυθμίσεων. Παρέχει τη δυνατότητα στο χρήστη να επιλέξει δικό του φάκελο με φωτογραφίες καθώς και να απενεργοποιήσει τη μουσική και τα ηχητικά εφέ (περισσότερα γι αυτό στη συνέχεια).

Επιπλέον έχουν υλοποιηθεί οι τάξεις Serializer, Responsive και Sound για τις διάφορες λειτουργίες του προγράμματος. Αναλυτικότερα:

- a) Serializer: Αυτή η τάξη είναι υπεύθυνη για την αποθήκευση και άντληση των δεδομένων των παικτών μέσω της τεχνικής του serialization.
- b) Responsive: Αυτή η τάξη έχει υλοποιηθεί για να διατηρείται το σωστό μέγεθος και η σωστή θέση όλων των controls ανεξάρτητα από τις αλλαγές στο μέγεθος της φόρμας. Όσες φόρμες και UserControl τη χρησιμοποιούν κάνουν implement το ειδικό interface IResponsive που έχει δημιουργηθεί γι αυτό το σκοπό. Συγκεκριμένα, λειτουργεί ως εξής: Στον constructor της responsive δίνεται σαν όρισμα μία φόρμα ή ένα user control ή ένα πάνελ και αποθηκεύεται το αρχικό μέγεθος αυτών καθώς και όλα τα controls που περιέχουν. Για την αναδιαστασιολόγηση των controls είναι απαραίτητο να είναι γνωστή η σχέση του μεγέθους του container και του μεγέθους και τοποθεσίας των controls. Γι αυτό έχει δημιουργηθεί η τάξη Ratio που αποθηκεύει αυτή τη σχέση για κάθε control. Τέλος, υπάρχει η μέθοδος ResizeUI η οποία παίρνει σαν όρισμα το νέο μέγεθος του container και αλλάζει το μέγεθος και την τοποθεσία των controls πολλαπλασιάζοντας το νέο μέγεθος του container με το ratio του μεγέθους και της τοποθεσίας του κάθε control.
- c) Sound: Αυτή η τάξη είναι υπεύθυνη για τη μουσική και τα ηχητικά εφέ. Για την αναπαραγωγή των αρχείων ήχου έχει χρησιμοποιηθεί η βιβλιοθήκη του windows media player.

2.2 Σκακιέρα

Το στήσιμο της εφαρμογής σκακιού βασίζεται στην εναλλαγή των panel. Με την έναρξη του παιχνιδιού βρισκόμαστε μπροστά στο πρώτο – menu panel, στο οποίο οι παίκτες ζητούνται να δώσουν μερικές πληροφορίες στο πρόγραμμα:

1. Τα ονόματα τους στα δύο textbox.
2. Να σημειώσουν πόσο χρόνο θα έχει ο κάθε παίκτης για τις κινήσεις του κατά την διάρκεια του παιχνιδιού.
3. Να επιλέξουν την σκακιέρα της αρεσκείας μέσα από τα τέσσερα προτεινόμενα σχέδια στα radiobuttons.

Εάν κάποιο από τα πιο πάνω δεν είναι σημειωμένο σωστά θα εμφανιστεί μήνυμα λάθους και το παιχνίδι δεν θα ξεκινήσει μέχρι να διορθωθούν.

Επιπλέον μέσα από το κουμπί «Player List» περνάμε στο panel “playerlistPanel” όπου μέσα από ένα DataGridView και με την κλάση DataReader εμφανίζονται τα δεδομένα της βάσης, δηλαδή τα ονόματα των προηγούμενων παικτών και τα χρονικά στοιχεία για το παιχνίδι τους.

```
1 reference
public class DataReader
{
    1 reference
    public void ReadData(string connectionString, DataGridView aDataGridView)
    {
        SQLiteConnection conn = new SQLiteConnection(connectionString);
        conn.Open();
        String querycommand = "Select * from Games";
        SQLiteCommand cmd = new SQLiteCommand(querycommand, conn);
        SQLiteDataAdapter sda = new SQLiteDataAdapter(cmd);
        DataTable dt = new DataTable();
        sda.Fill(dt);
        aDataGridView.AutoSizeColumns();
        aDataGridView.AutoSize = true;
        aDataGridView.DataSource = dt;
    }
}
```

Στην συνέχεια μέσα από το κουμπί “start” μεταφερόμαστε στην σκακιέρα. Εφόσον το παιχνίδι αρχίσει τα δεδομένα των παικτών καταχωρούνται στην βάση με την βοήθεια της κλάσης DataEntry και της συνάρτησης InsertData.

```
1 reference
public class DataEntry
{
    2 references
    public String Player1 { get; set; }
    2 references
    public String Player2 { get; set; }
    2 references
    public String Date { get; set; }
    1 reference
    public void InsertData (String player1, String player2, String connectionString)
    {
        Player1 = player1;
        Player2 = player2;
        Date = DateTime.Now.ToString();
        SQLiteConnection conn = new SQLiteConnection(connectionString);
        conn.Open();
        String insertQuery = "Insert into Games(player1,player2,date) values('" + Player1 + "','" + Player2 + "','" + Date + "')";
        SQLiteCommand mycommand = new SQLiteCommand(insertQuery, conn);
        mycommand.ExecuteNonQuery();
        conn.Close();
    }
}
```

Με την έναρξη του παιχνιδιού τα χρονόμετρα των δύο παιχτών ρυθμίζονται στον ορισμένο χρόνο και διορθώνονται τυχόν λάθη στον ορισμό τους(π.χ. αν έχουν δοθεί 100 δευτ. γίνονται 1λ. και 30 δευτ.). Μετά από κάθε κίνηση ο κάθε παίκτης χρειάζεται να πατούν το κουμπί “stop” ώστε να σταματήσει ο χρόνος τους και να ξεκινήσει η αντίστροφη μέτρηση για τον αντίπαλο.

Τα πιόνια κινούνται χάρη στα ομαδικά events “MouseDown” “MouseUp” “MouseMove”.



```
32 references
private void PawnW1_MouseDown(object sender, MouseEventArgs e)
{
    ((PictureBox)sender).BringToFront();
    point = e.Location;
    move = true;
}

32 references
private void PawnW1_MouseUp(object sender, MouseEventArgs e)
{
    move = false;
}

32 references
private void PawnW1_MouseMove(object sender, MouseEventArgs e)
{
    if (move)
    {
        ((PictureBox)sender).Location = new Point(((PictureBox)sender).Left + e.X - point.X, ((PictureBox)sender).Top + e.Y - point.Y);
    }
}
```

Κάθε παίκτης έχει την επιλογή “Give Up” σε περίπτωση που θέλει να τερματίσει το παιχνίδι, σε αυτή την περίπτωση οι χρόνοι σταματούν και εμφανίζεται το “Game over”. Το ίδιο συμβαίνει και εάν τελειώσει ένας από τους δύο αντίστροφους μετρητές. Εάν οι παίκτες θέλουν να ξεκινήσουν ένα ακόμα παιχνίδι μπορούν να πατήσουν το “Return” και να ανακατευθυνθούν στο menupanel και κρατήσουν τις προηγούμενες ρυθμίσεις είτε να κάνουν όποιες αλλαγές θελήσουν. Πατώντας το “Start” το παιχνίδι αρχίζει ξανά και τα πόνια έχουν επανατοποθετηθεί στις αρχικές τους θέσης χάρη στην κλάση Piece.

```
foreach (Piece piece in boardPanel.Controls.OfType<Piece>())
{
    piece.InitialLocation = piece.Location; //saves the location of the pieces
}

37 references
public class Piece : PictureBox
{
    34 references
    public Point InitialLocation { get; set; }
}
```

2.3 Διαχείριση Covid-19

Η εφαρμογή έχει υλοποιηθεί στο περιβάλλον του ASP .NET και ο σκοπός της είναι η βέλτιστη διαχείριση μιας βάσης δεδομένων μέσω των λειτουργιών της εισαγωγής, επεξεργασίας, αναζήτησης και διαγραφής μίας καταγραφής σε αυτήν (CRUD Operations). Το πρόγραμμα κάνει χρήση του κοντρόλ GridView(μια μορφή πίνακα για προβολή και διαχείριση δεδομένων) του ASP .NET και στα events αυτού καλούνται οι συναρτήσεις από την κλάση **DataHandler**. Επιπλέον, η κλάση **CovidCase** είναι υπεύθυνη για την καταγραφή των δεδομένων ενός νέου κρούσματος. Τέλος η εφαρμογή περιέχει και κάποια στατιστικά στο τέλος της σελίδας.

```
public class CovidCase
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string PhoneNumber { get; set; }
```



```
public string Gender { get; set; }  
public string Age { get; set; }  
public string Address { get; set; }  
public string Diseases { get; set; }  
public string Date { get; set; }
```

- `public CovidCase(string firstName, string lastName, string PhoneNumber, string gender, string age, string address, string diseases, string data)`

Ο κατασκευαστής (constructor) της κλάσης περιέχει σαν παραμέτρους όλα τα στοιχεία μιας καταγραφής(record) για την βάση δεδομένων. Στην δημιουργία αντικειμένου, δίνονται οι πληροφορίες στον κατασκευαστή(constructor) ο οποίος με την σειρά του τα αποθηκεύει στις μεταβλητές(property) της κλάσης για μελλοντική χρήση από άλλες μεθόδους.

```
}
```

```
public class DataHandler {
```

- `public SqlConnection ConnectToDatabase()`

Μέθοδος η οποία είναι υπεύθυνη για την σύνδεση με την βάση δεδομένων. Σε αυτήν τοποθετείται η συμβολοσειρά σύνδεσης(connection string). Η μέθοδος αυτή καλείται από τις υπόλοιπες μεθόδους της κλάσης DataHandler για να επιτευχθεί η σύνδεση με την βάση δεδομένων.

- `public DataTable ExecuteSqlCommand(string command, SqlConnection mysqlconnection)`

Μέθοδος η οποία είναι υπεύθυνη για την εκτέλεση SQL εντολών της μορφής SELECT. Με την σειρά της καλείται από άλλες μεθόδους της ίδιας κλάσης όπου χρειάζονται οι εντολές αυτής της μορφής.

- `public void InsertData(CovidCase newCovidCase)`

Μέθοδος η οποία είναι υπεύθυνη για την εισαγωγή δεδομένων στην βάση. Παίρνει ως όρισμα ένα αντικείμενο της κλάσης CovidCase το οποίο περιέχει τις πληροφορίες για κάθε στοιχείο που εισάγει ο χρήστης. Μέσω του αντικειμένου αυτού, αποκτά πρόσβαση στις μεταβλητές που περιέχουν τις πληροφορίες που θα εισαχθούν στην βάση.

- `public void BindData(GridView aGridView)`

Μέθοδος η οποία είναι υπεύθυνη για την ανανέωση του GridView. Επιλέγει όλα τα δεδομένα από την βάση δεδομένων και τα στέλνει ξανά πίσω στο GridView. Η συγκεκριμένη μέθοδος καλείται κάθε φορά στο τέλος κάθε



μεθόδου που επιτελεί κάποια λειτουργία(Insert,Search,Edit,Delete) για να έχουμε την εμφάνιση του αποτελέσματος της.

- `public void SearchData(GridView aGridView, TextBox searchkey)`

Μέθοδος η οποία είναι υπεύθυνη για την αναζήτηση μέσα στην βάση δεδομένων με βάση ένα `searchkey` το οποίο δόθηκε. Η αναζήτηση πραγματοποιείται με βάση το επώνυμο(LastName).

- `public void RowUpdateData(GridView aGridView, GridViewUpdateEventArgs e)`

Μέθοδος η οποία είναι υπεύθυνη για την επεξεργασία των δεδομένων κάθε γραμμής του GridView. Μετατρέπει τα fields σε TextBox στα οποία ο χρήστης μπορεί να συμπληρώσει τις αλλαγές που θέλει στην επιλεγμένη καταγραφή της βάσης δεδομένων.

- `public void RowDeleteData(GridView aGridView, GridViewDeleteEventArgs e)`

Μέθοδος η οποία είναι υπεύθυνη για την διαγραφή καταγραφών από την βάση δεδομένων. Η διαγραφή πραγματοποιείται μέσω του ID κάθε καταγραφής και γίνεται συνολική διαγραφή της επιλεγμένης γραμμής.
}

3 Επίδειξη των εφαρμογών

α) Παιχνίδι Μνήμης



Εικόνα 1: Main menu



Εικόνα 2: Options



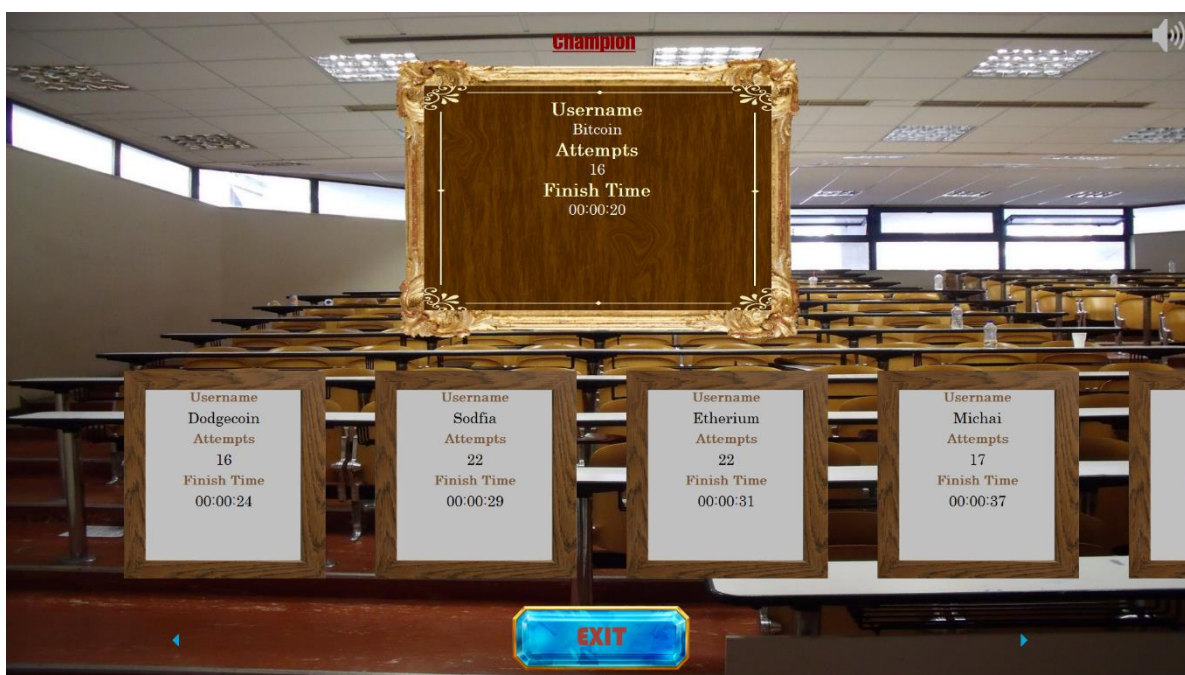
Εικόνα 3: Login screen



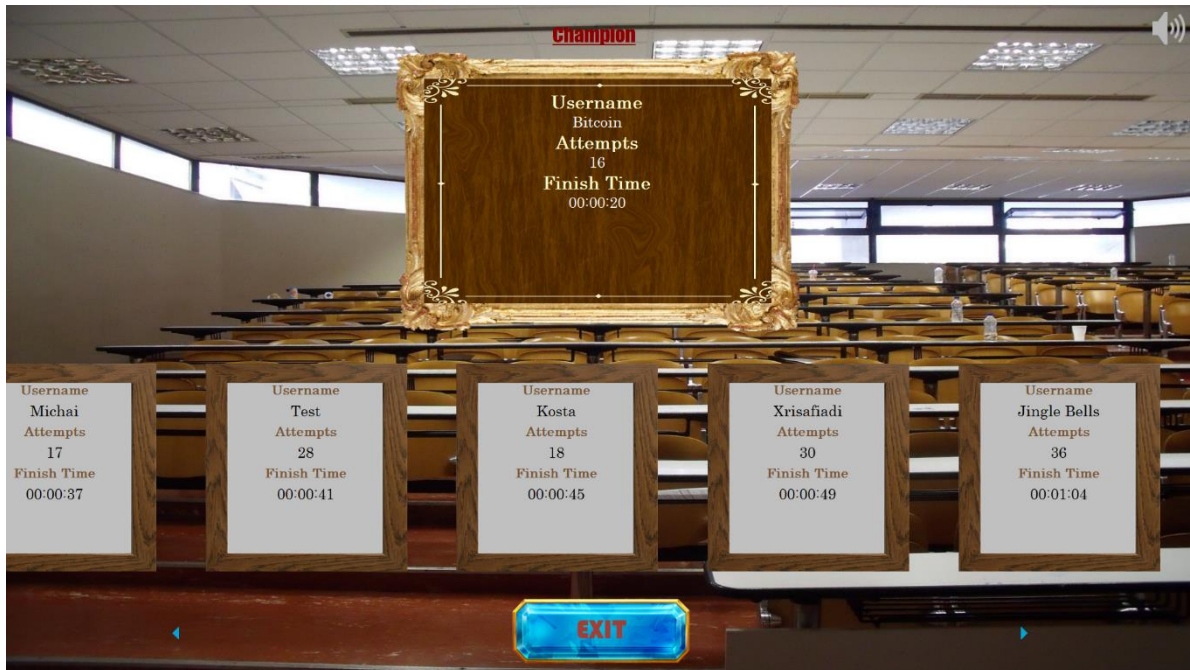
Εικόνα 4: Game, με όλες τις κάρτες κλειστές



Εικόνα 5: Game, με κάποιες ανοιχτές κάρτες

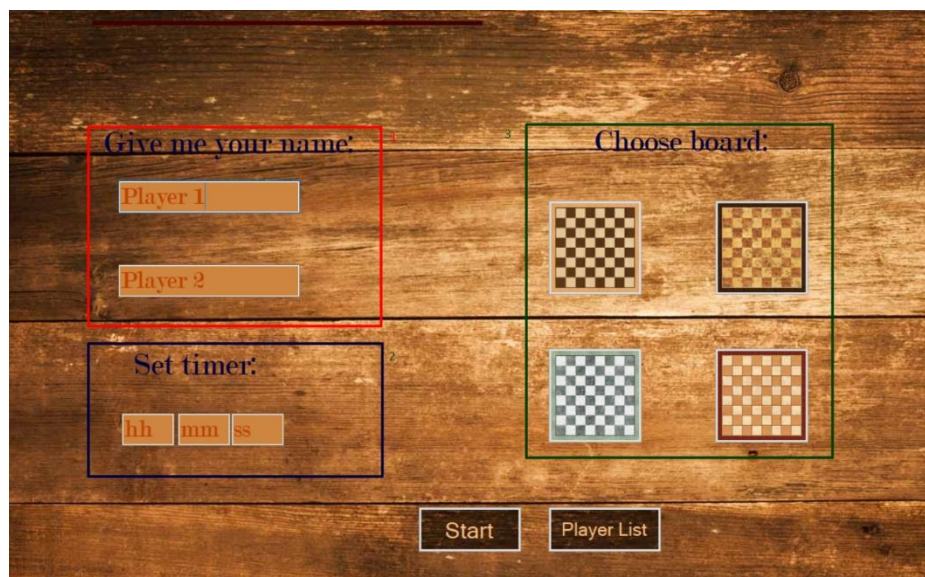


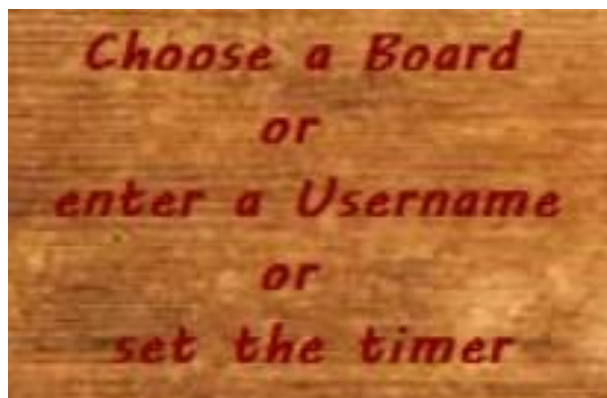
Εικόνα 6: Scoreboard, αρχική κατάσταση



Εικόνα 7: Scoreboard μετά απο μετακίνηση των καρτών

b) Σκακιέρα

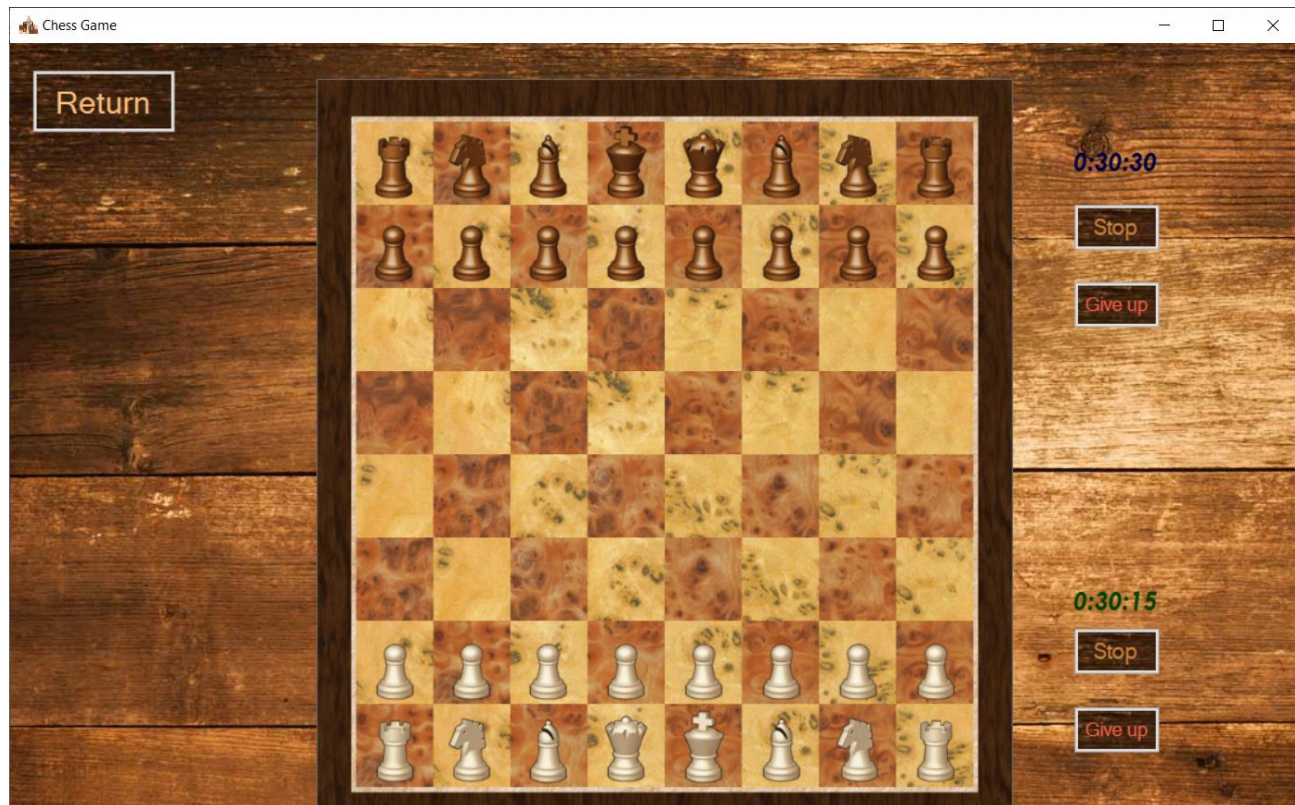




Εικόνα 8: Μήνυμα λάθους

	ID	player1	player2	date
▶	23	Kostas	Sofia	2/21/2021 5:59:...
	24	Kostas	Sofia	2/21/2021 5:59:...
	25	Aggelos	Giorgos	2/21/2021 6:00:...
	26	Aggelos	Giorgos	2/21/2021 6:00:...
	27	Kostas	Sofia	2/21/2021 6:02:...
	28	Michai	Player 2	21/2/2021 9:23:...
	29	Michai	Player 2	21/2/2021 9:24:...
	30	Haralampos	Theofania	22/2/2021 4:08:...
	44	Tester1	Tester2	2/22/2021 10:44...
	45	Tester1	Tester2	2/22/2021 10:45...
	46	Player 1	Player 2	2/22/2021 10:55...
	47	Mixalis	Sofia	2/22/2021 11:00...
	48	Mixalis	Sofia	2/22/2021 11:01...
	49	Kostas	Mixalis	2/22/2021 11:06...
	50	Kostas	Mixalis	2/22/2021 11:06...

Εικόνα 9: Απεικόνιση βάσης δεδομένων



Εικόνα 10: Σκακιέρα



Εικόνα 11: Τέλος παιχνιδιού



c) Διαχείριση Covid-19

Enter a covid case:

First Name:

Last Name:

Phone Number:

Gender:

☐ Male

☐ Female

Age:

Address:

Underlying Diseases:

Date:

Submit

Εικόνα 12 Εισαγωγή καταγραφής νέου κρούσματος

View, Edit and Search the database:

Enter a last name:

Search

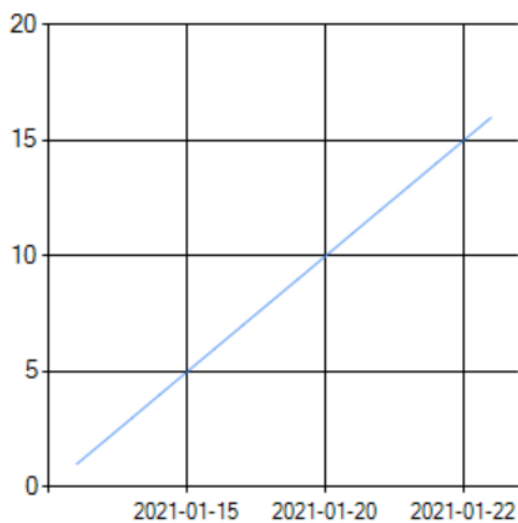
Id	FirstName	LastName	PhoneNumber	Gender	Age	Address	Deseases	Date	
1	Κωνσταντίνος	Καλογεροπουλος	6973901735	Male	19	Ανωγείων	-	2021-01-12	Edit Delete
2	Mixalis	Stylianidis	6973901734	Male	19	Testing Street 23	JAVA,C++,VAROUFAKIS	2021-01-13	Edit Delete
3	Sofia	Markopoulou	6973901733	Female	19	Testing Street 24	-	2021-01-14	Edit Delete
4	Aggeliki	Programmatismou	6973901732	Female	21	Testing Street 25	-	2021-01-15	Edit Delete
5	Ioanna	Programmatismou	6973901733	Female	25	Testing Street 26	-	2021-01-15	Edit Delete
6	Aggelos	Programmatismou	6973901733	Male	25	Testing Street 27	-	2021-01-16	Edit Delete
7	Nikos	Programmatismou	6973901444	Male	25	Testing Street 28	-	2021-01-17	Edit Delete
8	Theofaneia	Programmatismou	6973904334	Female	30	Testing Street 28	-	2021-01-18	Edit Delete
9	Areti	Programmatismou	6973903335	Female	24	Testing Street 29	-	2021-01-19	Edit Delete
10	Maria	Programmatismou	6973923444	Female	32	Testing Street 222	-	2021-01-20	Edit Delete

1 2

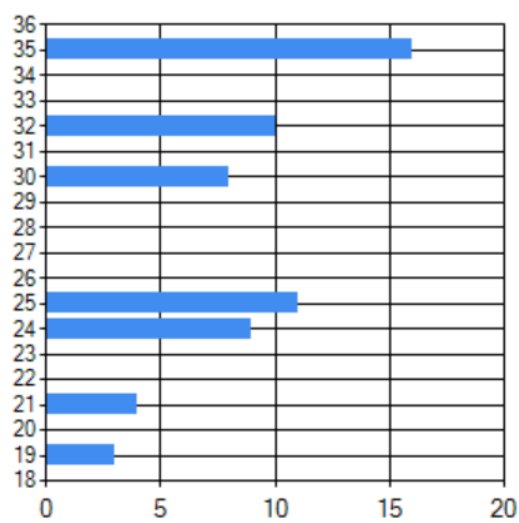
Εικόνα 13 Προβολή, επεξεργασία και αναζήτηση στην βάση δεδομένων

Statistics

Case's change/date



Number of covid cases/age



Εικόνα 14 Στατιστικά που προκύπτουν από τις καταγραφές της βάσης δεδομένων.