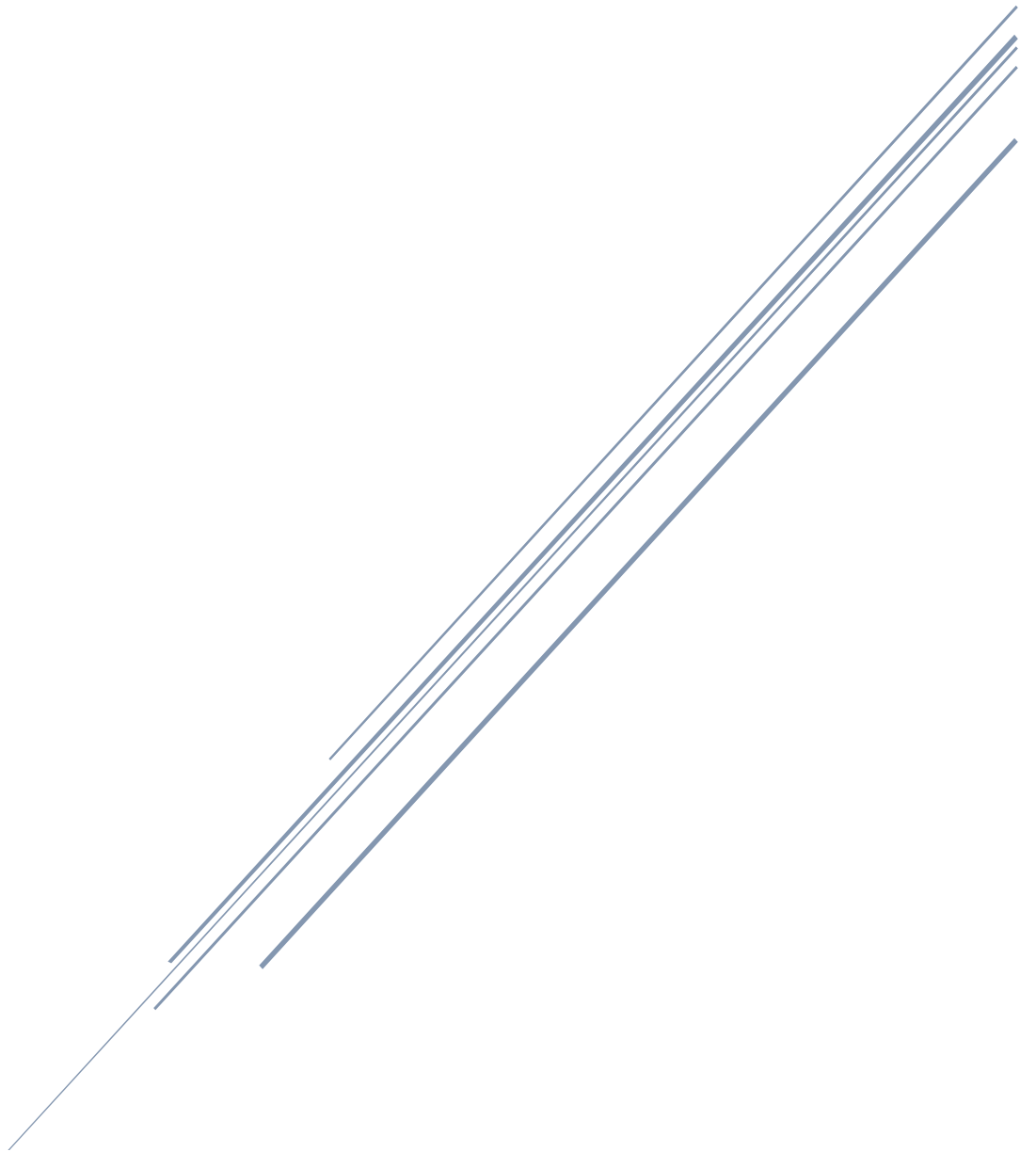


ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ (C#)

ΑΤΟΜΙΚΗ ΕΡΓΑΣΙΑ 2018-2019



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Δημήτρης Ματσαγγάνης
Π17068

Εκφώνηση ζητούμενης εργασίας

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΑΤΟΜΙΚΗ ΕΡΓΑΣΙΑ 2 ΜΟΝΑΔΩΝ 2018-2019

Ζητείται να υλοποιήσετε ένα παιχνίδι όπου ο στόχος του χρήστη είναι να “πετύχει” ένα γραφικό (της επιλογής σας) μέσα σε συγκεκριμένο χρόνο (π.χ. 1 λεπτό), ενώ το γραφικό αυτό θα μετακινείται «τυχαία» μέσα στην οθόνη, ανά μικρά χρονικά διαστήματα. Για κάθε «χτύπημα» του γραφικού, ο χρήστης θα παίρνει κάποιους πόντους/βαθμούς. Θα πρέπει να γίνεται αποθήκευση των πληροφοριών του παιχνιδιού (στοιχείων χρήστη, σκορ, κ.λπ.). Θα πρέπει το παιχνίδι να υποστηρίζει διαφορετικά επίπεδα δυσκολίας (τουλάχιστον 2). Επιπλέον, θα πρέπει να υποστηρίξετε τις λειτουργίες του μέγιστου σκορ, καθώς και μιας λίστας με τους καλύτερους παίκτες του παιχνιδιού. Παρακάτω σας δείχνω κάποιες ενδεικτικές οθόνες από τη χρήση του συγκεκριμένου παιχνιδιού (δεν είναι βέβαια αναγκαίο να προσπαθήσετε να φτιάξετε τις ίδιες οθόνες κι εσείς). Φυσικά, είστε ελεύθεροι να «εμπλουτίσετε» την εφαρμογή με επιπλέον δικές σας προσθήκες (που να έχουν νόημα για το παιχνίδι φυσικά). Εκτός από το project με τον κώδικα και το εκτελέσιμο αρχείο, θα πρέπει να φτιάξετε και μια σύντομη παρουσίαση με screenshots από την εφαρμογή σας, όπου θα αναφέρετε τι ακριβώς υλοποιήσατε (σε κάθε επίπεδο) και τι όχι, καθώς και τη «λογική» σας για την επίλυση του προβλήματος.

ΠΕΡΙΕΧΟΜΕΝΑ

1	Εισαγωγή της εργασίας.....	4
2	Βιβλιοθήκες, Αρχικοποίηση Μεταβλητών.	5
3	FORM1_LOAD (Φόρτωση Φόρμας 1).	5
4	Κινούμενο Αρχείο GIF (Mario, pictureBox1).	6
5	Start-Menu Button.	7
6	Ιστορικό Παικτών Ανά Επίπεδο Δυσκολίας.....	8
7	Επιλογή Δυσκολίας, Εξόδου Και Πληροφορίες Σχετικά Με Το Δημιουργό.....	9
8	Timer1_Tick (Κίνηση Εικόνας).	10
9	Timer2_Tick (Χρονόμετρο).	11
10	Pause button.....	13

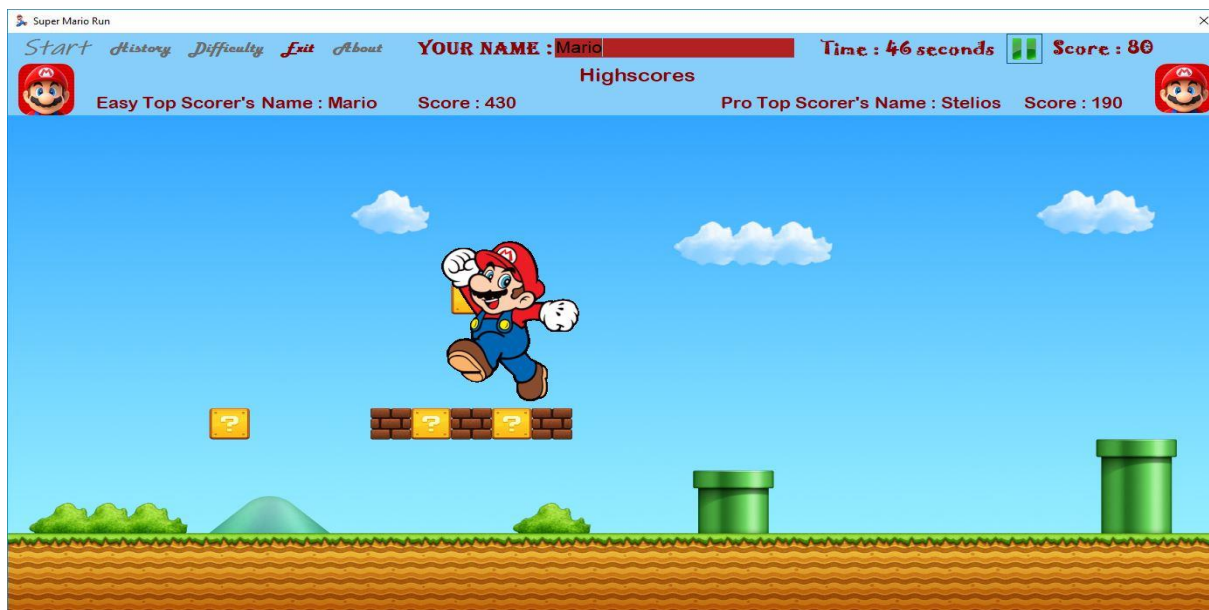
1 ΕΙΣΑΓΩΓΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Η παρούσα εργασία υλοποιεί ένα παιχνίδι κινουμένου στόχου, όπου ζητείται από το χρήστη να «πέτυχει» ένα κινούμενο γραφικό και για κάθε «χτύπημα» του κινουμένου στόχου ο χρήστης θα ανταμείβεται με πόντους, οι οποίοι διαφέρουν ανάλογα με το επίπεδο δυσκολίας που έχει επιλέξει.



Εικόνα Εφαρμογής 1: Η αρχική οθόνη του παιχνιδιού.

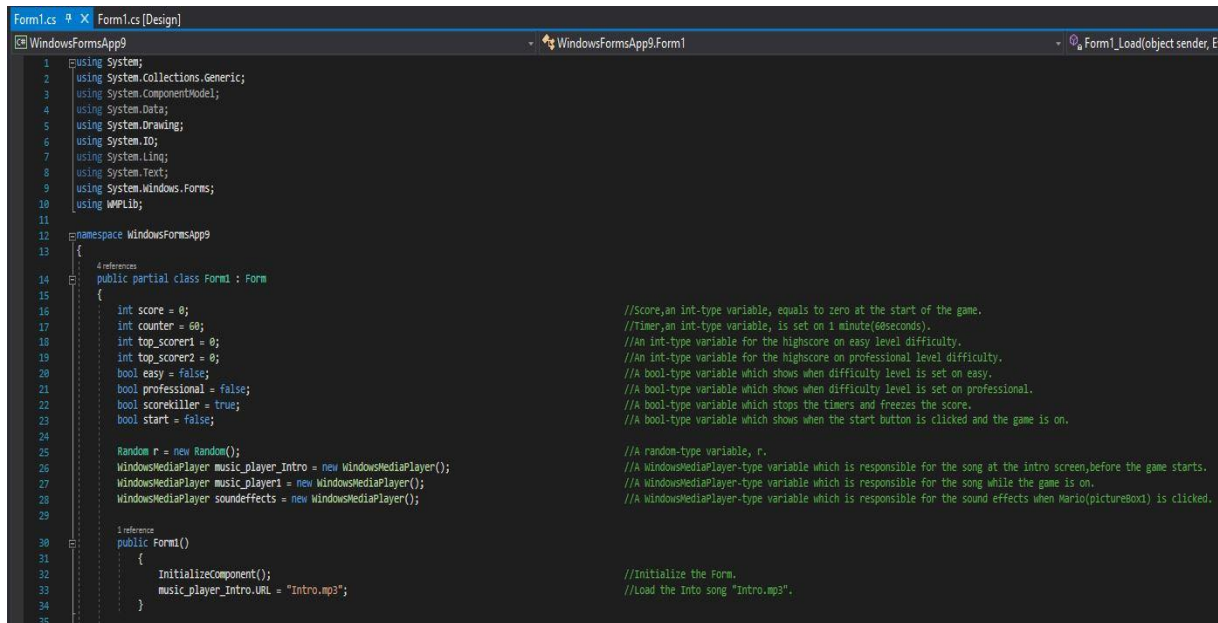
Η παρούσα εφαρμογή έχει ως κύριο πρωταγωνιστή τον Super Mario (το αρχείο GIF, pictureBox1), δύο επίπεδα δυσκολίας και τη δυνατότητα παύσης του παιχνιδιού και καταγραφής-αποθήκευσης των ονομάτων και των σκορ των παικτών και την ανάδειξη αυτού με το καλύτερο σκορ σε κάθε επίπεδο. Η εφαρμογή υλοποιήθηκε σε γλώσσα C# και με τη χρήση του εργαλείου Visual Studio . Στους αλγορίθμους επίλυσης της εργασίας έγινε χρήση γνώσεων και τεχνικών που παραδοθήκαν κατά τη διάρκεια του μάθημα Αντικειμενοστραφής Ανάπτυξη Εφαρμογών του 3^{ου} Εξάμηνου.



Εικόνα Εφαρμογής 2: Το παιχνίδι εν δράση.

2 ΒΙΒΛΙΟΘΗΚΕΣ, ΑΡΧΙΚΟΠΟΙΗΣΗ ΜΕΤΑΒΛΗΤΩΝ.

Στο πρόγραμμα έχουν συμπεριληφθεί οι απαραίτητες βιβλιοθήκες για την υλοποίησή του.

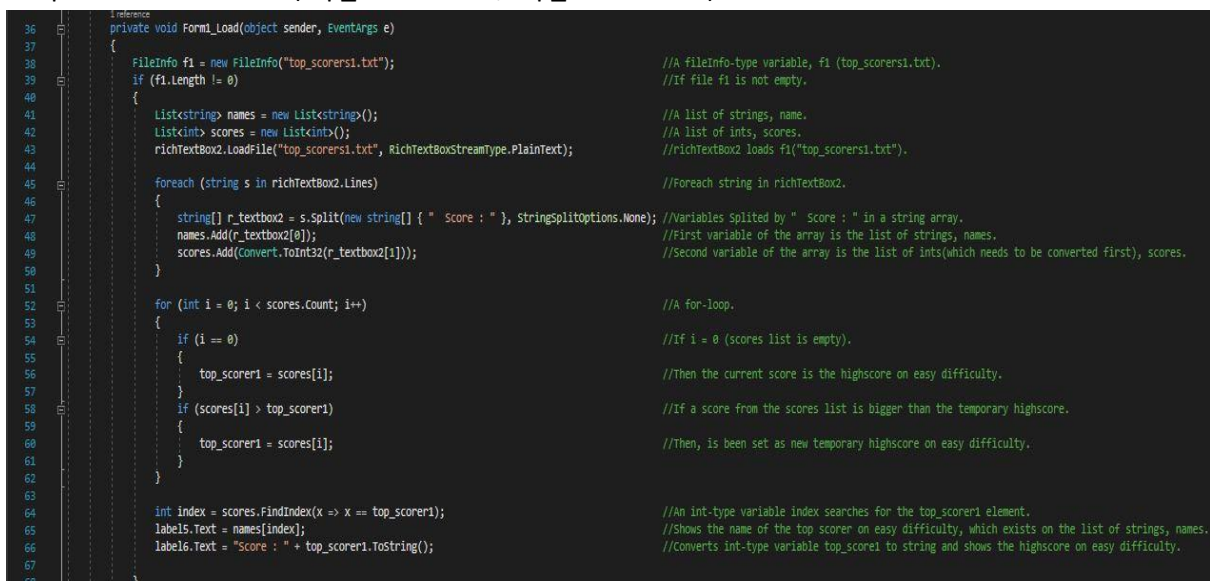


Εικόνα του κώδικα 1

Όπως, φαίνεται στην παραπάνω εικόνα γίνεται χρήση τεσσάρων ακεραίων μεταβλητών (int), τεσσάρων bool μεταβλητών, μίας random μεταβλητής και τριών WindowsMediaPlayer μεταβλητών. Έπειτα, αρχικοποιείται η φόρμα και γίνεται η φόρτωση του τραγουδιού της “Intro Screen” (Αρχικής Οθόνης) στη μεταβλητή “music_player_Intro”.

3 FORM1_LOAD (ΦΟΡΤΩΣΗ ΦΟΡΜΑΣ 1).

Με την φόρτωση της φόρμας, την ίδια χρονική στιγμή γίνεται και η ανάθεση σε δυο μεταβλητές τύπου Filename (f1 και f2), οι οποίες θα διαχειρίζονται τα αρχεία των παικτών με το μεγαλύτερο σκορ στο κάθε επίπεδο (top_scorers1.txt, top_scorers2.txt).



Εικόνα του κώδικα 2

Έπειτα, αν τα 2 προαναφερθέντα αρχεία δεν είναι κενά, τότε το όνομα και το σκορ του παίκτη με το μεγαλύτερο σκορ σε κάθε κατηγορία χωριστά θα φαίνεται στην οθόνη του παιχνιδιού. Στην περίπτωση που τα αρχεία είναι κενά τότε δεν θα εμφανιστεί κανένα όνομα και σκορ στην οθόνη του παιχνιδιού.

Κατά τη διάρκεια παραμονής του χρήστη στην αρχική οθόνη (χωρίς να πατήσει το Start και να εκκινήσει το παιχνίδι), το σκορ θα παραμένει όπως έχει ήδη οριστεί (δηλαδή 0) και θα ακούγεται η μουσική της αρχικής οθόνης ("Intro.mp3").

```

69 FileInfo f2 = new FileInfo("top_scorers2.txt"); //A FileInfo-type variable, f2 (top_scorers2.txt).
70 if (f2.Length != 0) //If file f2 is not empty.
71 {
72     List<string> names = new List<string>(); //A list of strings, name.
73     List<int> scores = new List<int>(); //A list of ints, scores.
74     richTextBox2.LoadFile("top_scorers2.txt", RichTextBoxStreamType.PlainText); //RichTextBox2 loads f2("top_scorers2.txt").
75
76     foreach (string s in richTextBox2.Lines) //foreach string in richTextBox2.
77     {
78         string[] r_textbox2 = s.Split(new string[] { " Score : " }, StringSplitOptions.None); //Variables Split by " Score : " in a string array.
79         names.Add(r_textbox2[0]); //First variable of the array is the list of strings, names.
80         scores.Add(Convert.ToInt32(r_textbox2[1])); //Second variable of the array is the list of ints (which needs to be converted first), scores.
81     }
82
83     for (int i = 0; i < scores.Count; i++) //A for-loop.
84     {
85         if (i == 0) //If i = 0 (scores list is empty).
86         {
87             top_scorer2 = scores[i]; //Then the current score is the highscore on professional difficulty.
88         }
89         if (scores[i] > top_scorer2) //If a score from the scores list is bigger than the temporary highscore.
90         {
91             top_scorer2 = scores[i]; //Then it is set as new temporary highscore on professional difficulty.
92         }
93     }
94
95     int index = scores.FindIndex(x => x == top_scorer2); //An int-type variable index searches for the top_scorer2 element.
96     label7.Text = names[index]; //Shows the name of the top scorer on professional difficulty which exists on the list of strings, names.
97     label8.Text = "Score : " + top_scorer2.ToString(); //converts int-type variable top_scorer2 to string and shows the highscore on professional difficulty.
98 }
99
100 scorekiller = true; //Score does not count while form is loaded, but the Start is not clicked (Intro Screen).
101 music_player.Intro.controls.play(); //The Intro song ("Intro.mp3") is playing while Start is not clicked (Intro Screen).
102 pictureBox1.ImageLocation = "Super_Mario.gif"; //pictureBox1 loads the "Super_Mario.gif".
103 SetStyle(ControlStyles.OptimizedDoubleBuffer, true); //DoubleBuffer is set on true.
104
105

```

Εικόνα του κώδικα 3

4 ΚΙΝΟΥΜΕΝΟ ΑΡΧΕΙΟ GIF (MARIO, PICTUREBOX1).

Στην παρούσα εργασία το κινούμενο αρχείο GIF βρίσκεται στο pictureBox1 και έχει ως πρωταγωνιστή το Super Mario.

```

106 1 reference
107 private void pictureBox1_Click(object sender, EventArgs e)
108 {
109     if (!scorekiller) //If the game is not paused or is in the Intro Screen.
110     {
111         string[] hit_sounds = new string[3] { "sound_effect1.mp3", "sound_effect2.mp3", "sound_effect3.mp3" }; //A string-type array of 3 songs("sound_effect1.mp3", "sound_effect2.mp3", "sound_effect3.mp3").
112
113         int i = r.Next(0, 3); //Int i is a random number choice between 1,2,3.
114         string sound_effect = hit_sounds[i]; //String sound_effect is a random choice from the string-type array hit_sounds.
115         soundeffects.controls.play(); //The sound_effects are playing at every click of the pictureBox1.
116         soundeffects.URL = sound_effect; //the random choice (string sound_effect) is loaded by the player (soundeffects).
117         label2.Text = "Score : " + score.ToString(); //Converts int-type variable score to string and shows the score.
118         if (easy) //If difficulty level is been set on easy difficulty.
119         {
120             score += 5; //At every click int-type variable score increases by 5.
121         }
122         else //If difficulty level is been set on professional difficulty.
123         {
124             score += 10; //At every click int-type variable score increases by 10.
125         }
126     }
127 }

```

Εικόνα του κώδικα 4

Επιπλέον, έχει ορισθεί ότι όσο το παιχνίδι είναι σταματημένο (paused) τότε, με την βοήθεια της μεταβλητής bool scorekiller, δεν θα μετράει το σκορ του παίκτη. Αυτό (δηλαδή : να μετράει το σκορ του παίκτη) γίνεται μόνο όταν η μεταβλητή scorekiller είναι απενεργοποιημένη (!scorekiller).

Τότε και μόνο τότε, ο παίκτης θα ανταμείβεται με πόντους, αναλόγως με το επίπεδο δυσκολίας που παίζει (5 πόντοι στο easy, 10 πόντοι στο professional), για κάθε του κλικ πάνω στο κινούμενο αρχείο GIF. Ακόμη, όταν κάνει κλικ πάνω στο pictureBox1 τότε θα ακούγεται ένας σύντομος ήχος, ο οποίος θα επιλέγεται τυχαία μέσα από την κλήση της τυχαίας μεταβλητής r (random-type variable r) και μιας λίστας, η οποία περιέχει σε μορφή string τα ονόματα των τριών πιθανών συντόμων ήχων. Η τελική

επιλογή θα μεταφέρεται στον soundeffects (WindowsMediaPlayer-type variable), ο οποίος θα παίζει τον επιλεγμένο σύντομο ήχο.

5 START-MENU BUTTON.

Με τον όρο Start-Menu Button εννοούμε το Start που εμπεριέχεται στο StripMenu. Έχει ορισθεί το παιχνίδι να μην ξεκινάει αν δεν έχει επιλεγεί από τον χρήστη το επίπεδο της δυσκολίας που επιθυμεί, σε κάθε τέτοια περίπτωση (δηλαδή: πάτημα του Start χωρίς να έχει επιλεγεί η δυσκολία του παιχνιδιού) θα εμφανίζεται ένα MessageBox που θα προειδοποιεί τον χρήστη και θα τον προτρέπει να επιλέξει ένα επίπεδο ώστε να αρχίσει το παιχνίδι.

```

128 private void startToolStripMenuItem_Click(object sender, EventArgs e)
129 {
130     if ((easy && !professional)) //If both "easy" and "professional" is false (i.e. No difficulty level is selected).
131     {
132         MessageBox.Show("Please select the level of difficulty you want.", "ERROR"); //A MessageBox pops up.
133     }
134     if (richTextBox1.Text == String.Empty) //If richTextBox1 is empty (i.e. No name from the user).
135     {
136         MessageBox.Show("You can't play without a name.", "ERROR"); //A MessageBox pops up.
137     }
138     else if (richTextBox1.Text != String.Empty && (easy || professional)) //If richTextBox1 is not empty and player has select a difficulty level (easy or professional).
139     {
140         if (easy) //If difficulty level is set on easy.
141         {
142             DialogResult dialogResult = MessageBox.Show("Try to catch Mario as many times as you can in order to get points.\nDifficulty has been set to Easy.\nYou have 60 seconds.", "Get ready?", MessageBoxButtons.OK);
143             //A MessageBox pops up with 1 possible answer (OK).
144             if (dialogResult == DialogResult.OK) //If answer is "OK".
145             {
146                 start = true; //Bool-type variable start is true.
147                 scorekiller = false; //Bool-type variable scorekiller is false (i.e. score counts).
148                 timer1.Enabled = true; //Timer1 enabled.
149                 timer2.Enabled = true; //Timer2 enabled.
150                 timer2.Interval = 1000; //At every 1000ms (= 1 second) Timer2 Ticks.
151                 music_player.Intro.controls.stop(); //The Intro song ("Intro.mp3") stops while Start is clicked (Play Screen).
152                 music_player1.Url = "Super Mario Bros.mp3"; //The Play song ("Super Mario Bros.mp3") loads at music_player1.
153                 music_player1.controls.play(); //The Play song ("Super Mario Bros.mp3") is playing when Start is clicked (Play Screen).
154                 pictureBox4.Visible = false; //pictureBox4 is not visible.
155                 pictureBox5.Visible = false; //pictureBox5 is not visible.
156                 pictureBox6.Visible = false; //pictureBox6 is not visible.
157                 pictureBox7.Visible = false; //pictureBox7 is not visible.
158                 richTextBox2.Visible = false; //richTextBox2 is not visible.
159                 startToolStripMenuItem.Enabled = false; //Start menu-button is not enabled.
160                 aboutToolStripMenuItem.Enabled = false; //About menu-button is not enabled.
161                 historyToolStripMenuItem.Enabled = false; //History menu-button is not enabled.
162                 label1.Text = "Time : " + counter.ToString() + " seconds"; //Converts int-type variable counter to string and shows the remaining time.
163             }
164         }
165     }

```

Εικόνα του κώδικα 5

Ακόμη, το παιχνίδι δεν θα ξεκινά εάν ο χρήστης δεν έχει συμπληρώσει το όνομα του στο κατάλληλο πεδίο (richTextBox1). Και σε αυτή την περίπτωση θα εμφανίζεται ένα MessageBox που θα τον προειδοποιεί και θα τον προτρέπει να συμπληρώσει το όνομα του.

Εάν έχουν τηρηθεί οι 2 αυτές προϋποθέσεις το παιχνίδι θα εμφανίζει ένα ακόμη MessageBox που θα προετοιμάζει τον χρήστη για την έναρξη του παιχνιδιού και με την απάντηση του (του MessageBox) το παιχνίδι θα ξεκινά στην ήδη επιλεγμένη από τον χρήστη δυσκολία.

```

166     else if (professional) //If difficulty level is set on easy.
167     {
168         DialogResult dialogResult = MessageBox.Show("Try to catch Mario as many times as you can in order to get points.\nBe careful because Mario is quicker now.\nDifficulty has been set to Professional.\nYou have 60 seconds.", "Are you ready?", MessageBoxButtons.OK);
169         //A MessageBox pops up with 1 possible answer (OK).
170         if (dialogResult == DialogResult.OK) //If answer is "OK".
171         {
172             start = true; //Bool-type variable start is true.
173             scorekiller = false; //Bool-type variable scorekiller is false (i.e. score counts).
174             timer1.Enabled = true; //Timer1 enabled.
175             timer2.Enabled = true; //Timer2 enabled.
176             timer2.Interval = 1000; //At every 1000ms (= 1 second) Timer2 Ticks.
177             music_player.Intro.controls.stop(); //The Intro song ("Intro.mp3") stops while Start is clicked (Play Screen).
178             music_player1.Url = "Super Mario Bros.mp3"; //The Play song ("Super Mario Bros.mp3") loads at music_player1.
179             music_player1.controls.play(); //The Play song ("Super Mario Bros.mp3") is playing when Start is clicked (Play Screen).
180             pictureBox4.Visible = false; //pictureBox4 is not visible.
181             pictureBox5.Visible = false; //pictureBox5 is not visible.
182             pictureBox6.Visible = false; //pictureBox6 is not visible.
183             pictureBox7.Visible = false; //pictureBox7 is not visible.
184             richTextBox2.Visible = false; //richTextBox2 is not visible.
185             startToolStripMenuItem.Enabled = false; //Start menu-button is not enabled.
186             aboutToolStripMenuItem.Enabled = false; //About menu-button is not enabled.
187             historyToolStripMenuItem.Enabled = false; //History menu-button is not enabled.
188             label1.Text = "Time : " + counter.ToString() + " seconds"; //Converts int-type variable counter to string and shows the remaining time.
189         }
190     }
191 }
192
193

```

Εικόνα του κώδικα 6

Κατά την έναρξη, η bool μεταβλητή start και αυτή του επιλεγμένου επιπέδου δυσκολίας γίνονται true, ενώ η scorekiller και η μη-επιλεγμένη δυσκολία γίνονται false, τα pictureBox4, pictureBox5, pictureBox6, pictureBox7 και richTextBox2 γίνονται μη ορατά (not visible), τα κουμπιά του StripMenu: Start, About, History μετατρέπονται σε ανενεργά κουμπιά και η μουσική αλλάζει σε αυτή του

παιχνιδιού ("Super Mario Bros.mp3"), σταματώντας την προηγούμενη του Intro Screen. Τέλος, ο χρόνος που απομένει θα γίνεται στο label1, το οποίο μετατρέπει την ακέραια (int) μεταβλητή counter σε string.

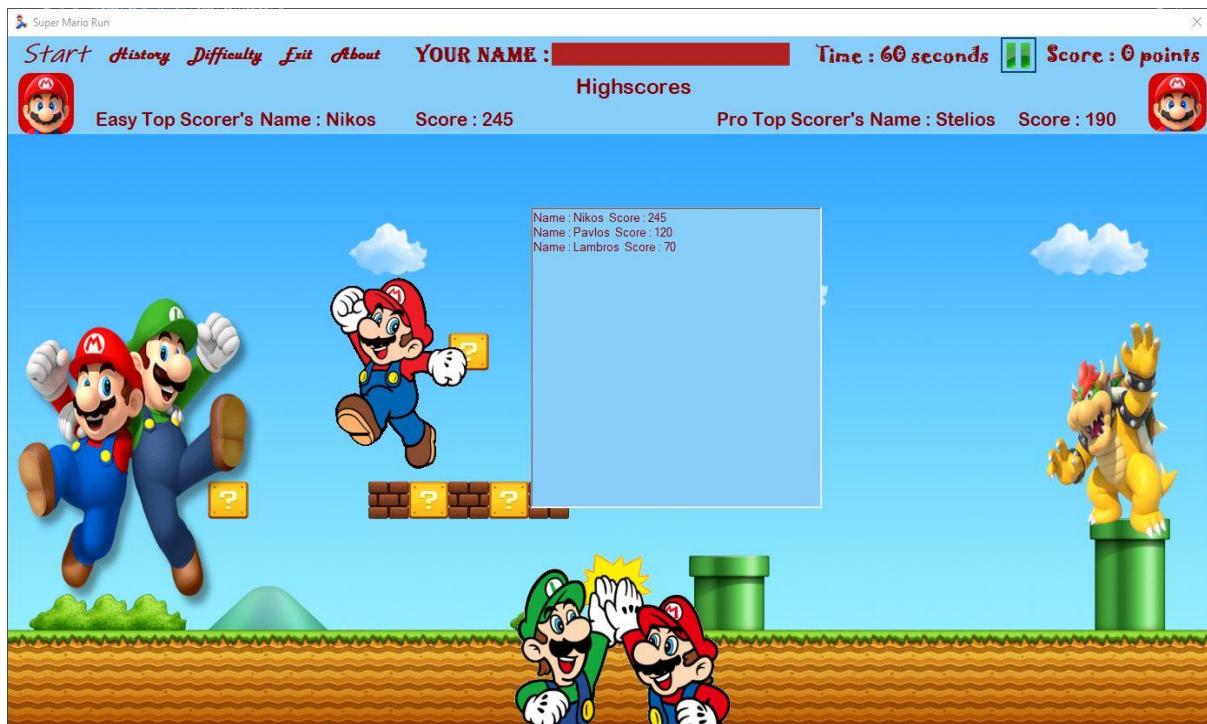
6 ΙΣΤΟΡΙΚΟ ΠΑΙΚΤΩΝ ΑΝΑ ΕΠΙΠΕΔΟ ΔΥΣΚΟΛΙΑΣ.

Όταν ο χρήστης θέλει να δει το ιστορικό των παικτών, τότε θα πάει μέσω του StripMenu στο κουμπί History και μετά θα πρέπει να διαλέξει το επίπεδο δυσκολίας για το οποίο θέλει να δει το ιστορικό.

```
194 private void easyToolStripMenuItem_Click(object sender, EventArgs e)
195 {
196     pictureBox4.Visible = false;           //pictureBox4 is not visible.
197     richTextBox2.Visible = true;          //richTextBox2 is visible (shows past players).
198     richTextBox2.LoadFile("top_scorers1.txt", RichTextBoxStreamType.PlainText); //richTextBox2 is loaded from "top_scorers1.txt" and shows past players.
199 }
200
201 private void professionalToolStripMenuItem_Click(object sender, EventArgs e)
202 {
203     pictureBox4.Visible = false;           //pictureBox4 is not visible.
204     richTextBox2.Visible = true;          //richTextBox2 is visible (shows past players).
205     richTextBox2.LoadFile("top_scorers2.txt", RichTextBoxStreamType.PlainText); //richTextBox2 is loaded from "top_scorers2.txt" and shows past players.
206 }
207
```

Εικόνα του κώδικα 7

Οι επιλογές είναι 2, όπως και τα επίπεδα δυσκολίας (easy και professional). Κατά την επιλογή του επιθυμητού επιπέδου το pictureBox4 γίνεται μη ορατό (not visible), ενώ το richTextBox2 γίνεται ορατό (visible) και φορτώνει τον κατάλληλο φάκελο ανάλογα με το επίπεδο δυσκολίας (Το top_scorers1.txt αν επιλέξει το ιστορικό του easy επιπέδου ή το top_scorers2.txt αν επιλέξει το ιστορικό του professional επιπέδου).



Εικόνα Εφαρμογής 3 :Το ιστορικό των παικτών.

7 ΕΠΙΛΟΓΗ ΔΥΣΚΟΛΙΑΣ, ΕΞΟΔΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΤΙΚΑ ΜΕ ΤΟ ΔΗΜΙΟΥΡΓΟ.

Όπως είχε αναφερθεί και προηγουμένως, ως απαραίτητη προϋπόθεση για την έναρξη του παιχνιδιού έχει τεθεί η επιλογή της επιθυμητής δυσκολίας, η οποία γίνεται μέσω του StripMenu και του κουμπιού Difficulty.

```

208 private void easyToolStripMenuItem_Click(object sender, EventArgs e)
209 {
210     DialogResult dialogResult = MessageBox.Show("Difficulty has been set to Easy.\nAre you ready to join Mario's world?", "Press Start to play", MessageBoxButtons.YesNo);
211     //A MessageBox pops up with 2 possible answers(Yes/No).
212     if (dialogResult == DialogResult.Yes) //If answer is "Yes".
213     {
214         easy = true; //Difficulty level is set on easy.
215         professional = false; //Bool-type variable professional is false (i.e.game on easy).
216         difficultyToolStripMenuItem.Enabled = false; //Difficulty menu-button is not enabled (i.e. game's difficulty cannot change while game is on).
217         pictureBox1.Size = new Size(300, 270); //pictureBox1 (Mario's) size changes.
218     }
219     else if (dialogResult == DialogResult.No) //If answer is "No", then do nothing.
220     {
221     }
222 }
223
224
225 private void professionalToolStripMenuItem_Click(object sender, EventArgs e)
226 {
227     DialogResult dialogResult = MessageBox.Show("Difficulty is set on Professional.\nAre you ready to join Mario's world?", "Press Start to play", MessageBoxButtons.YesNo);
228     //A MessageBox pops up with 2 possible answers(Yes/No).
229     if (dialogResult == DialogResult.Yes) //If answer is "Yes".
230     {
231         easy = false; //Bool-type variable easy is false (i.e.game on professional).
232         professional = true; //Difficulty level is set on professional.
233         difficultyToolStripMenuItem.Enabled = false; //Difficulty menu-button is not enabled (i.e. game's difficulty cannot change while game is on).
234         pictureBox1.Size = new Size(250, 250); //pictureBox1 (Mario's) size changes.
235     }
236     else if (dialogResult == DialogResult.No) //If answer is "No", then do nothing.
237     {
238     }
239 }
240
241
242 private void exitToolStripMenuItem_Click(object sender, EventArgs e)
243 {
244     Application.Exit(); // Exit from the Application.
245 }
246
247 private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
248 {
249     MessageBox.Show("This game was made by Dimitrios Matsanganis."); //A MessageBox pops up with editor's name.
250 }
251

```

Εικόνα του κώδικα 8

Έπειτα, υπάρχουν δύο επιλογές (δύο επίπεδα δυσκολίας) και αναλόγως με την επιλογή του χρήστη μεταβάλλεται και το μέγεθος του pictureBox1 (του Mario), μετά την επιλογή του επιπέδου το κουμπί του Difficulty στο StripMenu θα απενεργοποιηθεί μέχρι το τέλος του παιχνιδιού και σε περίπτωση που ο χρήστης αποφασίσει να ξαναπαίξει, τότε θα ξανά ενεργοποιηθεί ώστε, ο τελευταίος να επιλέξει ξανά το επίπεδο δυσκολίας που επιθυμεί για το νέο παιχνίδι.

Τα άλλα δύο κουμπιά του StripMenu, που δεν έχουν περιγράψει ακόμα, είναι το Exit, το οποίο με την επιλογή του τερματίζει την εφαρμογή και το About, το οποίο με την επιλογή του εμφανίζει μέσω ενός MessageBox κάποιες πληροφορίες για τον δημιουργό του παιχνιδιού.

8 TIMER1_Tick (ΚΙΝΗΣΗ ΕΙΚΟΝΑΣ).

Το timer1_Tick είναι υπεύθυνο για την κίνηση του pictureBox1 (του Mario) μέσα στην εφαρμογή. Ανάλογα, με το επίπεδο δυσκολίας η ταχύτητα μεταβολής της εικόνας αυξάνεται, δηλαδή ο χρόνος του timer1.Interval μειώνεται.

```

252 private void timer1_Tick(object sender, EventArgs e)
253 {
254     //Get the Width of the Form.
255     int W1 = this.Width;
256     //Get the Height of the Form.
257     int H1 = this.Height;
258     //If Difficult level is set on Easy.
259     if (easy)
260     {
261         timer1.Interval = 400;
262         //Every tick (Mario's movement), every 400ms.
263         //A sting-type array of 3 ints (-150, 0, 150).
264         int[] Random_Easy_Movement = new int[3] { -150, 0, 150 };
265         //Int-type variable, value1 is a random number choice.
266         int value1 = r.Next(0, 3);
267         //Int-type variable, value2 is a random number choice.
268         int value2 = r.Next(0, 3);
269         //Int-type variable, X_Easy_Movement is a random choice from the "Random_Easy_Movement" array.
270         int X_Easy_Movement = Random_Easy_Movement[value1];
271         //Int-type variable, Y_Easy_Movement is a random choice from the "Random_Easy_Movement" array.
272         int Y_Easy_Movement = Random_Easy_Movement[value2];
273         //Int-type variable, X is the move of the pictureBox1 on x-axis.
274         int X = pictureBox1.Location.X + X_Easy_Movement;
275         //Int-type variable, Y is the move of the pictureBox1 on y-axis.
276         int Y = pictureBox1.Location.Y + Y_Easy_Movement;
277         //New pictureBox1 (Mario) location.
278         pictureBox1.Location = new Point(X, Y);
279         //If X is bigger than the Width of the Form without Mario's width (pictureBox1,300px).
280         if (X > (W1 - 300))
281         {
282             //New pictureBox1 (Mario) location.
283             pictureBox1.Location = new Point(r.Next(300, (W1 - 300)), Y);
284         }
285         //If X is less than Mario's Width (pictureBox1,300px).
286         else if (X < 300)
287         {
288             //New pictureBox1 (Mario) location.
289             pictureBox1.Location = new Point(300, Y);
290         }
291         //If Y is bigger than the Height of the Form without Mario's Height (pictureBox1,270px).
292         if (Y > (H1 - 270))
293         {
294             //New pictureBox1 (Mario) location.
295             pictureBox1.Location = new Point(X, r.Next(250, (H1 - 270)));
296         }
297         //If Y is less than Mario's Height (pictureBox1,270px).
298         else if (Y < 270)
299         {
300             //New pictureBox1 (Mario) location.
301             pictureBox1.Location = new Point(X, 270);
302         }
303     }
304 }

```

Εικόνα του κώδικα 9

```

309 else if (professional)
310 {
311     //If Difficult level is set on Professional.
312     timer1.Interval = 350;
313     //Every tick (Mario's movement), every 350ms.
314     //A sting-type array of 5 ints (-250, -150, 0, 150, 250).
315     int[] Random_Professional_Movement = new int[5] { -250, -150, 0, 150, 250 };
316     //Int-type variable, value1 is a random number choice.
317     int value1 = r.Next(0, 5);
318     //Int-type variable, value2 is a random number choice.
319     int value2 = r.Next(0, 5);
320     //Int-type variable, X_Professional_Movement is a random choice from the "Random_Professional_Movement" array.
321     int X_Professional_Movement = Random_Professional_Movement[value1];
322     //Int-type variable, Y_Professional_Movement is a random choice from the "Random_Professional_Movement" array.
323     int Y_Professional_Movement = Random_Professional_Movement[value2];
324     //Int-type variable, X is the move of the pictureBox1 on x-axis.
325     int X = pictureBox1.Location.X + X_Professional_Movement;
326     //Int-type variable, Y is the move of the pictureBox1 on y-axis.
327     int Y = pictureBox1.Location.Y + Y_Professional_Movement;
328     //New pictureBox1 (Mario) location.
329     pictureBox1.Location = new Point(X, Y);
330     //If X is bigger than the Width of the Form without Mario's width (pictureBox1,250px).
331     if (X > (W1 - 250))
332     {
333         //New pictureBox1 (Mario) location.
334         pictureBox1.Location = new Point(r.Next(250, (W1 - 250)), Y);
335     }
336     //If X is less than Mario's Width (pictureBox1,250px).
337     else if (X < 250)
338     {
339         //New pictureBox1 (Mario) location.
340         pictureBox1.Location = new Point(250, Y);
341     }
342     //If Y is bigger than the Height of the Form without Mario's Height (pictureBox1,250px).
343     if (Y > (H1 - 250))
344     {
345         //New pictureBox1 (Mario) location.
346         pictureBox1.Location = new Point(X, r.Next(250, (H1 - 250)));
347     }
348     //If Y is less than Mario's Height (pictureBox1,250px).
349     else if (Y < 250)
350     {
351         //New pictureBox1 (Mario) location.
352         pictureBox1.Location = new Point(X, 250);
353     }
354 }
355 }

```

Εικόνα του κώδικα 10

Δύο ακεραίες μεταβλητές (int-type variables), W1 και H1 έχουν οριστεί ως το πλάτος και το ύψος της εφαρμογής (της φόρμας). Έπειτα, ορίζεται ένας πίνακας ακεραίων (με 3 ακεραίους στο easy επίπεδο και 5 ακεραίους στο professional επίπεδο), ο οποίος με την βοήθεια μιας τυχαίας μεταβλητής θα προσθέτει την τυχαία επιλεγμένη τιμή στην τοποθεσία που ήδη βρίσκεται το pictureBox1 (τόσο στον άξονα X όσο και στον άξονα Y) και θα ορίζει μια νέα θέση για το pictureBox1. Ωστόσο, για να παραμείνει το pictureBox1 μέσα στο πλαίσιο του παιχνιδιού (να μην βγαίνει από την οθόνη και να μην πηγαίνει κάτω από το StripMenu ή/και το Panel) τότε λαμβάνονται υπόψιν κάποιοι περιορισμοί, οι οποίοι σε περίπτωση που η τοποθεσία του pictureBox1 παραβεί αυτούς (κάποιες συντεταγμένες), τότε θα μεταφέρουν το pictureBox1 μέσα στα επιτρεπτά πλαίσια (τις συντεταγμένες του pictureBox1). Οι περιορισμοί αυτοί ενεργοποιούνται εάν: Η X συντεταγμένη (αντίστοιχα η συντεταγμένη Y) είναι μεγαλύτερη από το πλάτος (ύψος) της φόρμας αφού πρώτα έχει αφαιρεθεί το πλάτος (ύψος) του pictureBox1, τότε το X (αντίστοιχα το Y) θα μεταβληθεί και θα λάβει μια τυχαία

τιμή με ελάχιστη αυτή του πλάτους (ύψους) του pictureBox1 και μέγιστη αυτή του πλάτους (ύψους) της φόρμας χωρίς το πλάτος (ύψος) του pictureBox1. Ακόμη, εάν το πλάτος Χ (αντίστοιχα το ύψος Υ) είναι μικρότερο από το πλάτος (ύψος) του pictureBox1 (αν δεν ενεργοποιούνταν ο περιορισμός αυτός η εικόνα, ή ένα μέρος αυτής, θα ήταν εκτός των πλαισίων του παιχνιδιού), τότε το πλάτος Χ (αντίστοιχα το ύψος Υ) θα λάβει ως τιμή το πλάτος (ύψος) του pictureBox1, αναλόγως με το επίπεδο δυσκολίας πάντα.

9 TIMER2_TICK (ΧΡΟΝΟΜΕΤΡΟ).

Το timer2_Tick είναι υπεύθυνο για τον χρόνο διάρκειας του παιχνιδιού, για τη λήξη του παιχνιδιού, για την μεταβίβαση των στοιχείων των παικτών (όνομα και σκορ) στο κατάλληλο αρχείο αποθήκευσης, για να ελέγχει αν το σκορ που έκανε ο χρήστης είναι το καλύτερο σκορ στη κατηγορία του και να επαναφέρει τις ρυθμίσεις στην αρχική τους μορφή σε περίπτωση που ο χρήστης θέλει να ξανά παίξει.

```

323 private void timer2_Tick(object sender, EventArgs e)
324 {
325     counter--; //At every Timer2 Tick(1000ms), int-type variable counter is subtracted by 1.
326     label1.Text = "Time : " + counter.ToString() + " seconds"; //Converts int-type variable counter to string and shows the time remaining (label1).
327
328     if (counter == 0) //If int-type variable counter = 0 (Game is Over).
329     {
330         timer1.Stop(); //Stops Mario's movement (pictureBox1, timer1).
331         timer2.Stop(); //Stops clock's countdown (label1, timer2).
332
333         if (easy) //If difficulty level is been set on easy difficulty.
334         {
335             FileInfo f1 = new FileInfo("top_scorers1.txt"); //A FileInfo-type variable, f1 (top_scorers1.txt).
336             StreamWriter sw = new StreamWriter("top_scorers1.txt", true); //A StreamWriter-type variable, sw writes to (top_scorers1.txt).
337
338             if (f1.Length == 0) //If f1 (top_scorers1.txt) is empty.
339             {
340                 sw.WriteLine("Name : " + richTextBox1.Text + " Score : " + score); //Then, sw writes.
341             }
342             else if (f1.Length != 0) //If f1 (top_scorers1.txt) is not empty.
343             {
344                 sw.WriteLine(Environment.NewLine + "Name : " + richTextBox1.Text + " Score : " + score); //Then, sw writes in a new line.
345             }
346             sw.Close(); //StreamWriter-type variable, sw closes.
347             if (score > top_scorer1) //If score is bigger than the current highscore (top_scorer1).
348             {
349                 MessageBox.Show("NEW HIGHSCORE!"); //A MessageBox pops up.
350                 label5.Text = "Name : " + richTextBox1.Text; //label5's text changes and shows the name (richTextBox1.Text) of the new top scorer.
351                 label6.Text = " Score : " + score; //label6's text changes and shows the score (score) of the new top scorer.
352             }
353             else if (score < top_scorer1) //If score is less than the current highscore (top_scorer1), then do nothing.
354             {
355             }
356
357         }
358         else if (professional) //If difficulty level is been set on professional difficulty.
359         {
360             FileInfo f2 = new FileInfo("top_scorers2.txt"); //A FileInfo-type variable, f2 (top_scorers2.txt).
361             StreamWriter sw = new StreamWriter("top_scorers2.txt", true); //A StreamWriter-type variable, sw writes to (top_scorers2.txt).
362
363             if (f2.Length == 0) //If f2 (top_scorers2.txt) is empty.
364             {
365                 sw.WriteLine("Name : " + richTextBox1.Text + " Score : " + score); //Then, sw writes.
366             }
367             else if (f2.Length != 0) //If f2 (top_scorers2.txt) is not empty.
368             {
369                 sw.WriteLine(Environment.NewLine + "Name : " + richTextBox1.Text + " Score : " + score); //Then, sw writes in a new line.
370             }
371             sw.Close(); //StreamWriter-type variable, sw closes.
372             if (score > top_scorer2) //If score is bigger than the current highscore (top_scorer2).
373             {
374                 MessageBox.Show("NEW HIGHSCORE!"); //A MessageBox pops up.
375                 label7.Text = "Name : " + richTextBox1.Text; //label7's text changes and shows the name (richTextBox1.Text) of the new top scorer.
376                 label8.Text = " Score : " + score; //label8's text changes and shows the score (score) of the new top scorer.
377             }
378             else if (score < top_scorer2) //If score is less than the current highscore (top_scorer2), then do nothing.
379             {
380             }
381
382         }
383     }
384 }

```

Εικόνα του κώδικα 11

Αρχικά, μια ακέραια μεταβλητή (int type variable) counter θα μειώνεται ανά χτύπο του timer2 (timer2_Tick, ανά 1000ms./1sec.) ώστε, να λειτουργεί ορθολογικά το χρονόμετρο. Το χρονόμετρο θα φαίνεται στο label1. Με τη βοήθεια της μεταβλητής counter θα ελέγχεται εάν ο χρόνος του παιχνιδιού έχει τελειώσει (counter = 0), σε αυτή την περίπτωση τόσο ο timer1 όσο και ο timer2 θα σταματήσουν. Έπειτα, ανάλογα με το επίπεδο δυσκολίας που ήδη έχει επιλέξει ο χρήστης και με την βοήθεια μιας μεταβλητής τύπου StreamWriter, sw γίνεται αποθήκευση των πληροφοριών του νέου

παίκτη στο κατάλληλο αρχείο και η εμφάνιση των πληροφοριών αυτών αν το σκορ είναι το καλύτερο που έχει γίνει στο επιλεγμένο επίπεδο.

Στη συνέχεια, μετά το τέλος του παιχνιδιού (όταν counter = 0) θα εμφανίζεται ένα μήνυμα (MessageBox) με δυο πιθανές απαντήσεις (Yes/No) που καθορίζουν σε μεγάλο βαθμό το επόμενο στάδιο της εφαρμογής.

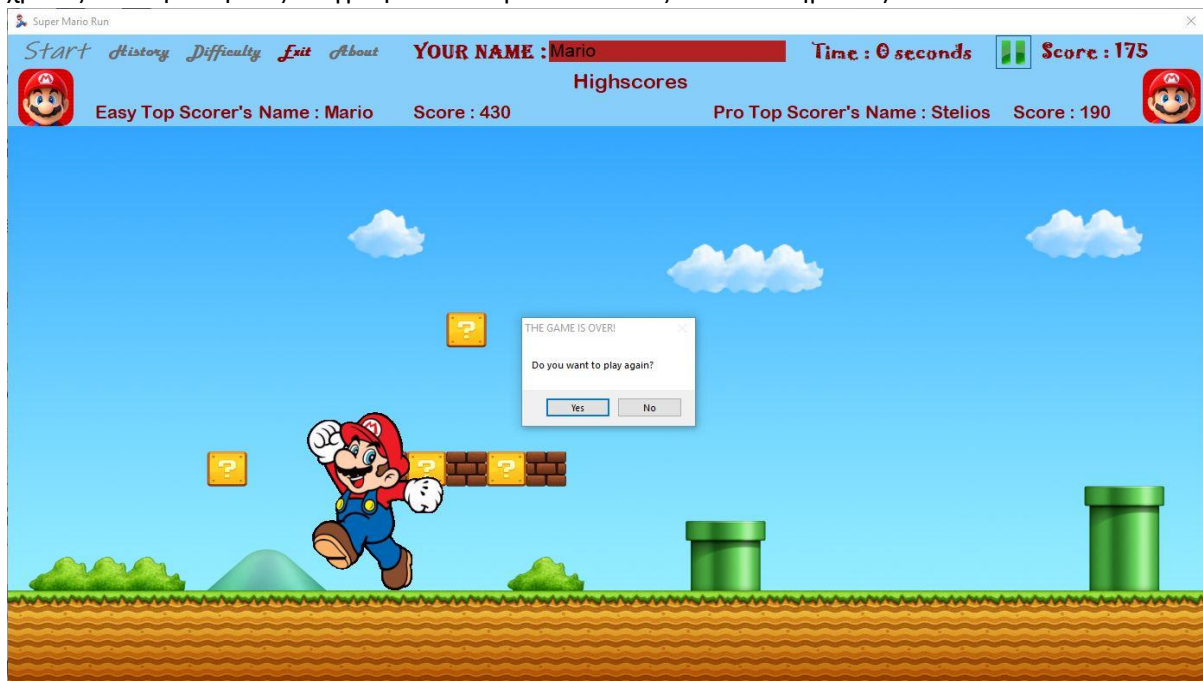
```

395 DialogResult dialogResult = MessageBox.Show("Do you want to play again?", "THE GAME IS OVER!", MessageBoxButtons.YesNo);
396 //A MessageBox pops up with 2 possible answers(Yes/No).
397
398 if (dialogResult == DialogResult.Yes) //If answer is "Yes" (i.e. Resets the Intro Screen).
399 {
400     score = 0; //Int-type variable, score resets to zero for a possible new game round.
401     counter = 60; //Int-type variable, counter resets to 60 for a possible new game round.
402     start = false; //Bool-type variable start is false, so is needed to click the Start for a new round.
403     easy = false; //Bool-type variable professional is false, so user need to select the difficulty level he/she wants in order to play again.
404     professional = false; //Bool-type variable easy is false, so user need to select the difficulty level he/she wants in order to play again.
405     scorekiller = true; //Bool-type variable scorekiller is true, so the score does not count.
406     label1.Text = "Time : " + counter.ToString() + " seconds"; //Converts int-type variable counter to string and shows the time.
407     label2.Text = "Score : " + score.ToString(); //Converts int-type variable score to string and shows the score.
408     music_player1.controls.stop(); //The Play song ("Super Mario Bros.mp3") stops while Start is not clicked (Intro Screen).
409     music_player1.intro.controls.play(); //The Intro song ("Intro.mp3") is playing while Start is not clicked (Intro Screen).
410     pictureBox1.Size = new Size(80, 270); //pictureBox1 (Mario) resets it's Intro Screen size.
411     pictureBox1.Location = new Point(225, 235); //pictureBox1 (Mario) resets it's Intro Screen location.
412     pictureBox4.Visible = true; //pictureBox4 is visible again.
413     pictureBox5.Visible = true; //pictureBox5 is visible again.
414     pictureBox6.Visible = true; //pictureBox6 is visible again.
415     pictureBox7.Visible = true; //pictureBox7 is visible again.
416     richTextBox2.Visible = false; //RichText2 is not visible.
417     startToolStripMenuItem.Enabled = true; //Start menu-button is enabled again.
418     aboutToolStripMenuItem.Enabled = true; //About menu-button is enabled again.
419     historyToolStripMenuItem.Enabled = true; //History menu-button is enabled again.
420     difficultyToolStripMenuItem.Enabled = true; //Difficulty menu-button is enabled again, so user can select the difficulty level he/she wants in order to play again.
421 }
422
423 else if (dialogResult == DialogResult.No) //If answer is "No".
424 {
425     Application.Exit(); //Exit from the Application.
426 }
427
428 }
429

```

Εικόνα του κώδικα 12

Αν η απάντηση στο MessageBox είναι "Yes", τότε η εφαρμογή θα επαναφέρεται στην αρχική της κατάσταση και οθόνη (Intro Screen), δηλαδή τα απενεργοποιημένα κουμπιά του StripMenu θα ενεργοποιηθούν ξανά, οι μη ορατές εικόνες (pictureBox4, pictureBox5, pictureBox6, pictureBox7) θα ξανά εμφανιστούν, οι bool μεταβλητές start, easy, professional θα γίνουν πάλι false, ενώ η scorekiller θα γίνει true ώστε, το σκορ να μην μετράει όταν βρισκόμαστε στην Αρχική Οθόνη, ακόμη η μουσική θα αλλάξει σε αυτήν της αρχικής οθόνης ("Intro.mp3"), το pictureBox1 θα πάρει το μέγεθος και τη θέση που είχε στην αρχική οθόνη και οι ακέραιες μεταβλητές score και counter θα λάβουν ξανά τις αρχικές τους τιμές (score = 0, counter = 60). Για να αρχίσει ξανά το παιχνίδι ο χρήστης αρκεί να επιλέξει το επίπεδο δυσκολίας που επιθυμεί και να πατήσει το κουμπί Start του StripMenu (δεν χρειάζεται δηλαδή να ξαναγράψει το όνομα του καθώς αυτό διατηρείται).



Εικόνα Εφαρμογής 4: Το παιχνίδι τελείωσε και το μήνυμα εμφανίζεται.

Αν η απάντηση στο MessageBox είναι "No", τότε η εφαρμογή θα τερματίζει αυτομάτως.

10 PAUSE BUTTON.

Ο σκοπός της κλήσης αυτού του κουμπιού είναι να λειτουργεί ως ένα κλασσικό κουμπί παύσης (Pause Button) με τη δυνατότητα εναλλαγής εικονιδίου.

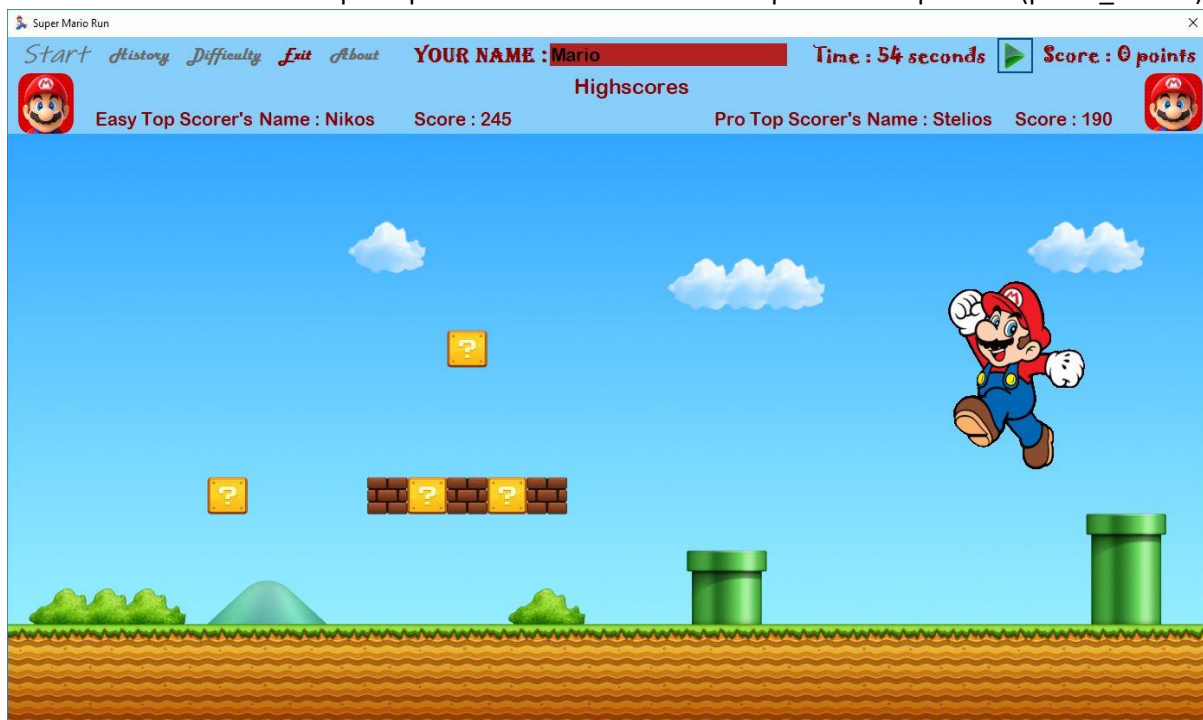
```

420 private void button1_Click(object sender, EventArgs e)
421 {
422     if (start)                                     //If Start button is clicked so the game is on.
423     {
424         if (!scorekiller)                         //If scorekiller = false, which means that the game is playing.
425         {
426             scorekiller = true;                   //On click the game is paused and the score is not counted (scorekiller = true).
427             timer1.Stop();                        //Stops Mario's movement (pictureBox1, timer1).
428             timer2.Stop();                        //Stops clock's countdown (label1, timer2).
429             button1.BackgroundImage = WindowsFormsApp9.Properties.Resources.play_button; //Changes the button1 background image to play_button.
430         }
431         else if (scorekiller)                     //If scorekiller = true, which means that the game is paused.
432         {
433             scorekiller = false;                  //On click the game is not paused and the score is counted (scorekiller = false).
434             timer1.Start();                       //Resumes Mario's movement (pictureBox1, timer1).
435             timer2.Start();                       //Resumes clock's countdown (label1, timer2).
436             button1.BackgroundImage = WindowsFormsApp9.Properties.Resources.pause_button; //Changes the button1 background image to pause_button.
437         }
438     }
439     else if (!start)                             //If Start is not clicked, then do nothing.
440     {
441     }
442 }
443 }
444 }
445 }
446 }

```

Εικόνα του κώδικα 13

Στο κουμπί αυτό έχει δοθεί η δυνατότητα να αλλάζει το εικονίδιο του (Background Image) αναλόγως με την προηγούμενη κατάσταση του. Πιο αναλυτικά, όταν το παιχνίδι δεν είναι σταματημένο (!scorekiller), τότε σε πιθανό κλικ από τον χρήστη θα σταματήσουν και οι δύο timers (timer1, timer2), η bool μεταβλητή scorekiller θα ενεργοποιηθεί (θα γίνει true, ώστε να μην μετράει το σκορ όταν το παιχνίδι έχει σταματήσει) και το εικονίδιο θα μετατραπεί στο κλασσικό κουμπί του play (play_button). Σε πιθανό κλικ του χρήστη ενώ το παιχνίδι είναι ήδη σταματημένο (paused, scorekiller = true), τότε οι timers θα συνεχίσουν από εκεί που είχαν μείνει, η bool μεταβλητή scorekiller θα απενεργοποιηθεί (θα γίνει false, ώστε να μετράει το σκορ όταν συνεχιστεί το παιχνίδι) και το εικονίδιο θα μετατραπεί στο κλασσικό κουμπί του pause (pause_button).



Εικόνα Εφαρμογής 5: Η παύση έχει ενεργοποιηθεί και το κουμπί έχει αλλάξει εικονίδιο.



Στη παραπάνω εικόνα φαίνεται το κουμπί του παύσης (pause_button) όταν το παιχνίδι είναι σταματημένο.