**עבודה מסכמת: הנדסת תוכנה ותכנות מובנה עצמים**

**הנדסת תעשייה וניהול, שנה ב' סמסטר ב'**

**מגישים:**

**ליטל יהודה -308134972-Lital5565**[**@gmail.com**](mailto:royon31@gmail.com)

**רועי כהן- 209544980 –** [**royon31@gmail.com**](mailto:royon31@gmail.com)

**הסבר על העובדה:**

בפתיחת הקובץ, מוכנסים לאוספים על פי סדר פרטי המידע מתוך קובץ infoDATA.txt

ופרטי שאלות הטריוויה gameDATA.txt ומתוך קבצי התמנות DIMAGES ו-QIMAGES .

כעת נפתח טופס להצגה עבור המשתמש, בטופס זה המשמש כטופס הראשי פתוח לתצוגה

תיבת טקסט למשתמש ופקד "login".

בתוך תיבת הטקסט צריך המשתמש לרשום כתובת מייל ובלחיצה על הפקד המערכת

בודקת האם המחרוזת שהוכנסה תואמת את כתיבת כתובת דוא"ל והאם המשתמש המוזן

לפני כתיבת סימן @ בכתובת הדוא"ל קיים ברשימת השחקנים במערכת(קריאה מקובץ

ובמידה והשם משתמש לא קיים אז כתיבת קובץ חדש באוסף שחקנים)

לאחר הבדיקה במידה ויצאה לא נכונה(כלומר הכתובת הוכנסה לא כראוי) יוצג למשתמש

הודעה שכותבת לו שהוא צריך לנסות שוב.

במידה והבדיקה הצליחה מוסתרות בפני המשתמש שני הפקדים הראשונים ומוצגות בפניו 3

אפשרויות שהם החלקים של הממשק:

**חלק א- חלק לימודי**

בלחיצה על פקד זה מהמסך הראשי נפתח למשתמש טופס נוסף בטופס יופיעו 12 קטעי

למידה הנבחרים באקראיות מתוך אוסף פרטי המידע שקראנו מהקובץ המתאים, 12 פרטי

המידע נבחרים ומוסיפים למערך וכך יכול המשתמש בעזרת פקדים לדלג קדימה ואחורה

במערך של 12 קטעי המידע.

**חלק ב- חלק משחק טריוויה**

בלחיצה על פקד זה מהמסך הראשי נבחרות למשתמש באופן רנדומלי מתוך אוסף השאלות

10 שאלות לתוך מערך המייצג את המשחק הספציפי של אותו משתמש.

למשחק הטריוויה יש 4 סוגי שאלות שעבור כל סוג נפתחים הפקדים המתאימים לו שלבסוף

בוחר המשמש בתשובה ומסמן את ה radio Butten המייצג את התשובה הנכונה עבורו

סוגי השאלות הם: נכון\לא נכון, שאלות עם 3 תשובות כאשר תשובה אחת נכונה, נכון/לא

נכון כאשר השאלה נתמכת בתמונה ושאלות עם 3 אפשרויות שונות, שהאפשרויות הן

תמונות ותמונה אחת היא הנכונה.

במידה והמשתמש צדק בתשובתו הוא מקבל נקודה המופיעה לו במעלה המסך וסה"כ יכול

המשתמש לקבל עד 10 נקודות.

בנוסף מעל כמות הנקודות שלו מופיע למשתמש טיימר המייצג כמה זמן(בשניות) עבר

מתחילת המשחק.

בסוף המשחק כאשר המשתמש לוחץ על פקד "סיום" נפתח למשתמש מסך המראה לו את

הסטטיסטיקה של המשחק שעשה(ניקוד, זמן, כמות השאלות שהצליח ושטעה),

סטטיסטיקה זו נוספת לקובץ של אותו שחקן לתוך קובץ txt שבו שמורים פרטי המשחק

שעשה. לאחר מכן יכול המשתמש ללחוץ על פקד שמעביר אותו חזרה לחלון הראשי.

**חלק ג- עדכון תוכן**

בלחיצה על פקד זה נפתח למשתמש טופס נוסף ובו הוא יכול לבחור ב2 אפשרויות:

אפשרות 1: הוספת פריט מידע- המשתמש צריך להוסיף נושא ותוכן לקטע מידע ועל פי

רצונו גם תמונה. בלחיצה על "הוסף" במידה כל הפרטים הוכנסו ייכתב למשתמש בהערה

שהתוכן הוסף ואכן הקטע מידע שהוסיף יוסף לרשימת קטעי המידע בקובץ txt ולאוסף קטעי

המידע בממשק. במידה ולא הכניס את אחד השדות יופיע למשתמש הערה שאומרת שעליו

למלא את כל השדות עבור הוספה.

אפשרות 2: הוספת שאלה- יפתח למשתמש טופס שבו יבחר איזה סוג של שאלה הוא רוצה

להוסיף לפי הנושאים שהוצגו בחלק הטריוויה. לפי בחירתו יוצג למשתמש איזה שדות עליו

למלא ובדומה להוספת פריט מידע , השאלה שהוסיף תוסף לקובץ הtxt המתאים לפי סוג

השאלה שבחר ובנוסף השאלה תתווסף לאוסף השאלות בממשק.

**פלטי הקוד:**

**:DataItemמחלקת**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Drawing;

using System.Threading.Tasks;

namespace GeometricShapes

{

public class DataItem : IComparable // מחלקת בסיס של פריט מידע

{

protected int number;

protected string topic;

protected string content;

public DataItem(int number, string topic, string content)

{

this.Number = number;

this.Topic = topic;

this.Content = content;

}

public string Topic { get => topic; set => topic = value; }

public string Content { get => content; set => content = value; }

public int Number

{

get => number;

set

{

if (value < 0)

{

throw new ArgumentException("The item number canot be negative");

}

number = value;

}

}

public int CompareTo(Object obj) //מתודת השוואה בין 2 פריטים ע"י שדה Content

{

DataItem item = (DataItem)obj;

return content.CompareTo(item.content);

}

public override string ToString()

{

return "\r\n" + this.number + ";" + this.topic + ";" + this.content;

}

}

}

**:** **DataItemWImage מחלקת**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Drawing;

using System.Threading.Tasks;

namespace GeometricShapes

{

public class DataItemWImage : DataItem // מחלקה נגזרת של מחלקת פריט מידע  
 {

protected Image image;

public DataItemWImage(int number, string topic, string content, Image image) : base(number, topic, content)

{

this.image = image;

}

public Image Image { get => image; set => image = value; }

}

}

**:** **Questionמחלקת**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Drawing;

using System.Threading.Tasks;

namespace GeometricShapes

{

public enum Qtype //משתנה של סוג השאלה

{

TrueFalse = 0, Multiple, TrueFalseWImage, MultipleWImages

}

public class Questions :IComparable //מחלקת שאלות ותשובות עם 4 בנאים שונים עבור כל סוג  
 {

protected int qnumber;

protected string question;

protected Image imageQuestion;

protected string correctAnswer;

protected string incorrectAnswer1;

protected string incorrectAnswer2;

protected Image imageCorrectAnswer;

protected Image imageIncorrectAnswer1;

protected Image imageIncorrectAnswer2;

public Questions(int qnumber, Qtype qtype, string question, string correctAnswer, string incorrectAnswer1) //בנאי לשאלת אמת או שקר  
 {

this.qnumber = qnumber;

this.Qtype = qtype;

this.question = question;

this.correctAnswer = correctAnswer;

this.incorrectAnswer1 = incorrectAnswer1;

}

public Questions(int qnumber, Qtype qtype, string question, string correctAnswer, string incorrectAnswer1, string incorrectAnswer2) //בנאי לשאלת ריבוי בחירה

{

this.qnumber = qnumber;

this.Qtype = qtype;

this.question = question;

this.correctAnswer = correctAnswer;

this.incorrectAnswer1 = incorrectAnswer1;

this.incorrectAnswer2 = incorrectAnswer2;

}

public Questions(int qnumber, Qtype qtype, string question, string correctAnswer, string incorrectAnswer1, Image imageQuestion) //בנאי לשאלת אמת או שקר עם תמונה  
 {

this.qnumber = qnumber;

this.Qtype = qtype;

this.question = question;

this.imageQuestion = imageQuestion;

this.correctAnswer = correctAnswer;

this.incorrectAnswer1 = incorrectAnswer1;

}

public Questions(int qnumber, Qtype qtype, string question, Image ImagecorrectAnswer, Image ImageincorrectAnswer1, Image ImageincorrectAnswer2) //בנאי לשאלת ריבוי בחירה עם תמונה  
 {

this.qnumber = qnumber;

this.Qtype = qtype;

this.question = question;

this.ImageCorrectAnswer = ImagecorrectAnswer;

this.ImageIncorrectAnswer1 = ImageincorrectAnswer1;

this.ImageIncorrectAnswer2 = ImageincorrectAnswer2;

}

public Qtype Qtype { get ; set ; }

public int Qnumber { get => qnumber; set => qnumber = value; }

public Image ImageQuestion { get => imageQuestion; set => imageQuestion = value; }

public string Question { get => question; set => question = value; }

public string CorrectAnswer { get => correctAnswer; set => correctAnswer = value; }

public string IncorrectAnswer1 { get => incorrectAnswer1; set => incorrectAnswer1 = value; }

public string IncorrectAnswer2 { get => incorrectAnswer2; set => incorrectAnswer2 = value; }

public Image ImageCorrectAnswer { get => imageCorrectAnswer; set => imageCorrectAnswer = value; }

public Image ImageIncorrectAnswer1 { get => imageIncorrectAnswer1; set => imageIncorrectAnswer1 = value; }

public Image ImageIncorrectAnswer2 { get => imageIncorrectAnswer2; set => imageIncorrectAnswer2 = value; }

public int CompareTo(Object obj) //מתודת השוואה בין שאלה שקיימת במערך לבין שאלה חדשה  
 {

Questions item = (Questions)obj;

return question.CompareTo(item.question);

}

public List<string> TrueOrFalseAnswer() //מתודת עזר שמחזירה רשימת עזר של תשובות אמת או שקר  
 {

List<string> TrueOrFalse = new List<string>();

TrueOrFalse.Add(correctAnswer);

TrueOrFalse.Add(incorrectAnswer1);

return TrueOrFalse;

}

public List<string> MultipleAnswers()//מתודת עזר שמחזירה רשימת עזר של ריבוי תשובות   
 {

List<string> Multiple = new List<string>();

Multiple.Add(correctAnswer);

Multiple.Add(incorrectAnswer1);

Multiple.Add(incorrectAnswer2);

return Multiple;

}

public List<Image> MultipleImages()//מתודת עזר שמחזירה רשימת עזר של ריבוי תשובות עם תמונות

{

List<Image> MultipleImages = new List<Image>();

MultipleImages.Add(imageCorrectAnswer);

MultipleImages.Add(imageIncorrectAnswer1);

MultipleImages.Add(imageIncorrectAnswer2);

return MultipleImages;

}

}

}

**:** **Player מחלקת**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace GeometricShapes

{

class Player //מחלקת משתמש

{

protected readonly string nickName;  
 protected string userEmail;

protected string date;

protected string path;

public Player(string userEmail) // בנאי שיוצר קובץ סטטיסטיקת שחקן ספציפי בתאריך ספציפי   
 {

this.UserEmail = userEmail;

nickName = userEmail.Split('@')[0];

Date = (DateTime.Now.ToString("yyyy-MM-dd'T'HH':'mm':'ss")).Replace(':', '\_') + ".txt"; //שמירת שם הקובץ עפ"י datetime

if (Directory.Exists("DATA\\PLAYERS\\" + nickName) == false) // בדיקה האם תקייה של שם המשתמש הנוכחי קיימת, במידה ולא- נוצרת תקייה חדשה שם שמו

{

Directory.CreateDirectory("DATA\\PLAYERS\\" + nickName);

}

Path = $"DATA\\PLAYERS\\{nickName}\\" + nickName + "\_" + Date;

File.Create(Path).Close();

}

public string NickName => nickName;

public string UserEmail { get => userEmail; set => userEmail = value; }

public string Date { get => date; set => date = value; }

public string Path { get => path; set => path = value; }

}

}

**:** **Globals מחלקת**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Drawing;

namespace GeometricShapes

{

public static class Globals //מחלקת עזר סטאטית המכילה מערך של שאלות ותשובות ורשימה של פרטי מידע  
 {

public static ArrayList gameData; // מערך של שאלות ותשובות למשחק טריוויה

public static List<DataItem> infoData; //רשימה של פרטי מידע

public static void LoadGameData() // קריאה מתוך קובץ השאלות והתשובות והוספת נתונים לצערך   
 {

gameData = new ArrayList();

StreamReader stream = new StreamReader("DATA\\gameData.txt");

while (!stream.EndOfStream)

{

string row = stream.ReadLine();

string[] dataItem = row.Split(';');

if (int.Parse(dataItem[1]) == 0) // הוספת שאלה ותשובות מסוג 0 למערך

{

Questions item0 = new Questions(int.Parse(dataItem[0]), (Qtype)int.Parse(dataItem[1]), dataItem[2], dataItem[3], dataItem[4]);

gameData.Add(item0);

}

else if (int.Parse(dataItem[1]) == 1) // הוספת שאלה ותשובות מסוג 1 למערך  
 {

Questions item1 = new Questions(int.Parse(dataItem[0]), (Qtype)int.Parse(dataItem[1]), dataItem[2], dataItem[3], dataItem[4], dataItem[5]);

gameData.Add(item1);

}

else if (int.Parse(dataItem[1]) == 2) // הוספת שאלה כולל תמונה ותשובות מסוג 2 למערך

{

Image image = Image.FromFile(Directory.GetCurrentDirectory() + @"\DATA\QIMAGES\" + dataItem[5]);

Questions item2 = new Questions(int.Parse(dataItem[0]), (Qtype)int.Parse(dataItem[1]), dataItem[2], dataItem[3], dataItem[4], image);

gameData.Add(item2);

}

else if (int.Parse(dataItem[1]) == 3) // הוספת שאלה ותשובות מסוג 3 למערך

{

Image image1 = Image.FromFile(Directory.GetCurrentDirectory() + @"\DATA\QIMAGES\" + dataItem[3]);

Image image2 = Image.FromFile(Directory.GetCurrentDirectory() + @"\DATA\QIMAGES\" + dataItem[4]);

Image image3 = Image.FromFile(Directory.GetCurrentDirectory() + @"\DATA\QIMAGES\" + dataItem[5]);

Questions item3 = new Questions(int.Parse(dataItem[0]), (Qtype)int.Parse(dataItem[1]), dataItem[2], image1, image2, image3);

gameData.Add(item3);

}

}

stream.Close();

}

public static void LoadInfoData() // קריאה מתוך קובץ המידע והוספת הנתונים לרשימת  
 {

infoData = new List<DataItem>();

StreamReader stream = new StreamReader("DATA\\infoData.txt");

while (!stream.EndOfStream)

{

string row = stream.ReadLine();

string[] dataItem = row.Split(';');

if (dataItem.Length == 3) //הוספת פריט לרשימה ללא תמונה

{

DataItem item = new DataItem(int.Parse(dataItem[0]), dataItem[1], dataItem[2]);

infoData.Add(item);

}

else //הוספת פריט לרשימה עם תמונה

{

Image image = Image.FromFile(Directory.GetCurrentDirectory() + @"\DATA\DIMAGES\" + dataItem[3]);

DataItemWImage item = new DataItemWImage(int.Parse(dataItem[0]), dataItem[1], dataItem[2], image);

infoData.Add(item);

}

}

stream.Close();

}

}

}

**:** **Program מחלקת**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

**:MainForm טופס**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Text.RegularExpressions;

namespace GeometricShapes

{

public partial class MainForm : Form //טופס ראשי לבחירת הפעולה הרצוייה ואימות מייל המשתמש

{

public static string userEmail; //משתנה עזר סטאטי ששומר את הכתובת מייל התקינה של המשתמש

public MainForm()

{

InitializeComponent();

this.ClientSize = new System.Drawing.Size(725, 515);

Globals.LoadGameData(); //טעינת קובץ השאלות והתשובות ועדכון המערך

Globals.LoadInfoData(); //טעינת קובץ המידע מחדש ועדכון הרשימה

}

private void btnTrivia\_Click(object sender, EventArgs e) //מעבר לטופס משחק הטריוויה

{

TriviaForm trivia = new TriviaForm();

trivia.ShowDialog();

}

private void btnStudy\_Click(object sender, EventArgs e) //מעבר לטופס למידה

{

StudyForm study = new StudyForm();

study.ShowDialog();

}

private void btnUpdate\_Click(object sender, EventArgs e) //מעבר לטופס הוספת מידע או שאלה

{

UpdateForm update = new UpdateForm();

update.ShowDialog();

}

private void MainForm\_FormClosing(object sender, FormClosingEventArgs e) //סגירת תוכנית

{

Application.Exit();

}

private void btnLogin\_Click(object sender, EventArgs e) //כפתור אישור כניסה למערכת

{

userEmail = txtLogin.Text;

bool res = CheckEmail(userEmail);

if (res == true)

{

lblLogin.Visible = false;

txtLogin.Visible = false;

btnLogin.Visible = false;

btnTrivia.Visible = true;

btnStudy.Visible = true;

btnUpdate.Visible = true;

}

else

MessageBox.Show("The email in not valid, Please try again", "Error", MessageBoxButtons.OK);

}

private bool CheckEmail(string userEmail) //מתודת אימות כתובת מייל

{

Regex email = new Regex(@"[0-9a-zA-Z]+@[0-9a-zA-Z]+\.[0-9a-zA-Z]+");

if (email.IsMatch(userEmail))

return true;

return false;

}

}

}

**:** **StudyForm טופס**

using System;

using System.IO;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

public partial class StudyForm : Form //טופס חלק לימודי

{

int selectedRow = 0;

List<DataItem> dataRand = new List<DataItem>(); //רשימת עזר עם 12 פריטים רנדומליים

Random rand = new Random();

public StudyForm()

{

InitializeComponent();

btnBackwards.Visible = false;

}

private void StudyForm\_Load(object sender, EventArgs e)

{

try

{

Globals.LoadInfoData(); //טעינת קובץ המידע מחדש ועדכון הרשימה

int counter = 0;

while(counter <12) //הוספת 12 פריטים רנדומליים לרשימת העזר

{

int index = rand.Next(0, Globals.infoData.Count() - 1);

if (!dataRand.Contains(Globals.infoData[index])) //כל עוד השאלות לא חוזרת על עצמן

{

dataRand.Add(Globals.infoData[index]);

counter++;

}

}

LoadItem();

}

catch (FileNotFoundException)

{

MessageBox.Show("File Not Found");

}

catch (OutOfMemoryException)

{

MessageBox.Show("Insufficient memory to allocate a buffer for the returned string Exception");

}

catch (Exception m)

{

MessageBox.Show("Exception" + m.Message);

}

}

private void btnDone\_Click(object sender, EventArgs e) //הודעה על סיום פעולת הלמידה

{

MessageBox.Show("Well Done! Now you are ready");

this.Close();

}

private void btnBackwards\_Click(object sender, EventArgs e) //מתודת דפדוף אחורה

{

btnForward.Visible = true;

selectedRow--;

if (selectedRow == 0)

btnBackwards.Visible = false;

LoadItem();

}

private void btnForward\_Click(object sender, EventArgs e) //מתודת דפדוף קדימה

{

btnBackwards.Visible = true;

selectedRow++;

if (selectedRow == 11)

btnForward.Visible = false;

LoadItem();

}

private void LoadItem() //טעינת הפריט מרשימת העזר והצגתו למשתמש

{

lblIndex.Text = "" + (selectedRow + 1) + "/12";

txtShowContact.Text = dataRand[selectedRow].Content;

lblShowTopic.Text = dataRand[selectedRow].Topic;

if (dataRand[selectedRow] is DataItemWImage) //אופציית הצגת תמונה כשמדובר בפריט עם תמונה

{

pbxShowPic.Image = ((DataItemWImage)dataRand[selectedRow]).Image;

pbxShowPic.Visible = true;

}

else

pbxShowPic.Visible = false;

}

private void btnBack\_Click(object sender, EventArgs e) //כפתור חזרה לתפריט הראשי

{

this.Close();

}

}

}

**:** **TriviaForm טופס**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

public partial class TriviaForm : Form //טופס משחק הטריוויה

{

int selectedRow = 0, tick = 0, score;

Player player;

List<Questions> dataRand = new List<Questions>(); //רשימת עזר עם 10 שאלות רנדומליות

Random rand = new Random();

Questions current;

public TriviaForm()

{

InitializeComponent();

score = 0;

this.ClientSize = new System.Drawing.Size(730, 580);

rbtAnswer1.Checked = false;

rbtAnswer2.Checked = false;

rbtAnswer3.Checked = false;

}

private void btnBack\_Click(object sender, EventArgs e) //כפתור חזרה לתפריט הקודם

{

DialogResult dr = MessageBox.Show("Are you sure?","The game didn't save",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dr == System.Windows.Forms.DialogResult.Yes)

{

TriviaTimer.Stop();

this.Close();

}

}

private void TriviaForm\_Load(object sender, EventArgs e) //טעינת קובץ השאלות והתשובות מחדש, עדכון המערך ויצירת רשימת עזר

{

try

{

Globals.LoadGameData(); //טעינת קובץ השאלות והתשובות מחדש ועדכון המערך

int counter = 0;

while (counter < 10) //הוספת 10 שאלות ותשובות רנדומליות לרשימת העזר

{

int index = rand.Next(0, Globals.gameData.Count - 1);

if (!dataRand.Contains(Globals.gameData[index]))

{

dataRand.Add((Questions)Globals.gameData[index]);

counter++;

}

}

LoadQuestion();

}

catch (FileNotFoundException)

{

MessageBox.Show("File Not Found");

}

catch (OutOfMemoryException)

{

MessageBox.Show("Insufficient memory to allocate a buffer for the returned string Exception");

}

catch (Exception m)

{

MessageBox.Show("Exception" + m.Message);

}

TriviaTimer.Start();

tick = 0;

}

private void btnSaveGame\_Click(object sender, EventArgs e) //שמירת פרטי המשחק, כתיבה לקובץ הסטטיסטיקה ויצירת שחקן עפ"י מייל האימות הראשוני

{

player= new Player(MainForm.userEmail);

TriviaTimer.Stop();

writeToFile();

this.Close();

this.Hide();

FormResults results = new FormResults(player.Path); //מעבר לתוצאות המשחק

results.ShowDialog();

}

private void TriviaTimer\_Tick(object sender, EventArgs e) //מתודה שמציגה לשחקן את הזמן שהוא נמצא במשחק

{

tick++;

txtTimer.Text = "" + tick;

}

private void btnForward\_Click(object sender, EventArgs e) //מתודת דפדוף קדימה ובדיקת התשובה שנבחרה

{

selectedRow++;

if (selectedRow == 9)

btnForward.Visible = false;

if (CheckAnswers() == true) //בדיקה האם התשובה שנבחרה היא הנכונה, במידה וכן הניקוד עולה בנקודה אחת

{

score++;

txtScore.Text = score.ToString();

}

Clear();

LoadQuestion();

}

private void LoadQuestion() //טעינת השאלה והתשובות מרשימת העזר על פי סוג השאלה והצגתן למשתמש

{

lblIndex.Text = "" + (selectedRow + 1) + "/10";

current = dataRand[selectedRow];

txtQuestion.Text = current.Question;

switch (dataRand[selectedRow].Qtype)

{

case Qtype.TrueFalse:

int res = Rendom2Answers(); //הגרלת סדר התשובות

rbtAnswer1.Text = current.TrueOrFalseAnswer()[res];

rbtAnswer2.Text = current.TrueOrFalseAnswer()[1-res];

break;

case Qtype.TrueFalseWImage:

pbxPicQuestion.Image = current.ImageQuestion;

pbxPicQuestion.Visible = true;

res = Rendom2Answers(); //הגרלת סדר התשובות

rbtAnswer1.Text = current.TrueOrFalseAnswer()[res];

rbtAnswer2.Text = current.TrueOrFalseAnswer()[1 - res];

break;

case Qtype.Multiple:

int first = Rendom3Answers(); //הגרלת סדר התשובות

int secnd = rand.Next(3);

while (secnd == first)

secnd = rand.Next(3);

rbtAnswer1.Text = current.MultipleAnswers()[first];

rbtAnswer2.Text = current.MultipleAnswers()[secnd];

rbtAnswer3.Text = current.MultipleAnswers()[3 - first - secnd];

break;

case Qtype.MultipleWImages:

pcbAnswer1.Visible = true;

pcbAnswer2.Visible = true;

pcbAnswer3.Visible = true;

first = Rendom3Answers(); //הגרלת סדר התשובות

secnd = rand.Next(3);

while (secnd == first)

secnd = rand.Next(3);

pcbAnswer1.Image = current.MultipleImages()[first];

pcbAnswer2.Image = current.MultipleImages()[secnd];

pcbAnswer3.Image = current.MultipleImages()[3-first-secnd];

break;

}

}

private int Rendom2Answers() // פונקציית עזר שמגרילה סדר של 2 תשובות להצגה

{

rbtAnswer1.Visible = true;

rbtAnswer2.Visible = true;

int randNum = rand.Next(2);

return randNum;

}

private int Rendom3Answers() // פונקציית עזר שמגרילה סדר של 3 תשובות להצגה

{

rbtAnswer1.Visible = true;

rbtAnswer2.Visible = true;

rbtAnswer3.Visible = true;

int randNum = rand.Next(3);

return randNum;

}

private bool CheckAnswers() // פונקציית עזר שמחזירה אמת במידה והתשובה שבחר השחקן זהה לתשובה הנכונה

{

switch (current.Qtype)

{

case Qtype.TrueFalseWImage:

if (rbtAnswer1.Checked && current.CorrectAnswer == rbtAnswer1.Text)

return true;

else if (rbtAnswer2.Checked && current.CorrectAnswer == rbtAnswer2.Text)

return true;

break;

case Qtype.TrueFalse:

if (rbtAnswer1.Checked && current.CorrectAnswer == rbtAnswer1.Text)

return true;

else if (rbtAnswer2.Checked && current.CorrectAnswer == rbtAnswer2.Text)

return true;

break;

case Qtype.Multiple:

if (rbtAnswer1.Checked && current.CorrectAnswer == rbtAnswer1.Text)

return true;

else if (rbtAnswer2.Checked && current.CorrectAnswer == rbtAnswer2.Text)

return true;

else if (rbtAnswer3.Checked && current.CorrectAnswer == rbtAnswer3.Text)

return true;

break;

case Qtype.MultipleWImages:

if (rbtAnswer1.Checked && current.ImageCorrectAnswer == pcbAnswer1.Image)

return true;

else if (rbtAnswer2.Checked && current.ImageCorrectAnswer == pcbAnswer2.Image)

return true;

else if (rbtAnswer3.Checked && current.ImageCorrectAnswer == pcbAnswer3.Image)

return true;

break;

}

return false;

}

private void Clear() //מתודה שמאפסת את הפקדים למצב התחלתי

{

rbtAnswer1.Visible = false;

rbtAnswer2.Visible = false;

rbtAnswer3.Visible = false;

rbtAnswer1.Text = "";

rbtAnswer2.Text = "";

rbtAnswer3.Text = "";

pbxPicQuestion.Visible = false;

pcbAnswer1.Visible = false;

pcbAnswer2.Visible = false;

pcbAnswer3.Visible = false;

rbtAnswer1.Checked = false;

rbtAnswer2.Checked = false;

rbtAnswer3.Checked = false;

}

public void writeToFile() //כתיבה לתוך קובץ הסטטיסטיקה של השחקן עבור משחק ספציפי

{

File.AppendAllText(player.Path, "User Email: " + player.UserEmail + "\r\nHow long the game took: " + txtTimer.Text

+ " sec\r\ncorrect answers: " + score + "\r\nincorrect answers: " + (10 - score));

}

}

}

**:** **FormResults טופס**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

public partial class FormResults : Form //טופס הצגת תוצאות המשחק

{

string fileName;

public FormResults(string str) //בנאי מלא המקבל מחרוזת עם שם התקייה שנוצרה עבור אותו שחקן

{

InitializeComponent();

fileName = str;

}

private void btnBack\_Click(object sender, EventArgs e) //כפתור סגירת הטופס

{

this.Close();

}

private void FormResults\_Load(object sender, EventArgs e)

{

StreamReader stream = new StreamReader(fileName); //קריאה מתוך הקובץ הרלוונטי שנוצר עובר משחק ספציפי ועבור שחקן ספציפי

txtResult.Text= "\r\n\r\n" +stream.ReadToEnd(); //הצגת התוצאות לשחקן

stream.Close();

}

}

}

**:** **UpdateForm טופס**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

public partial class UpdateForm : Form //טופס בחירת נתונים שאותם רוצים להוסיף למאגר

{

public UpdateForm()

{

InitializeComponent();

}

private void btnAdditem\_Click(object sender, EventArgs e) //מעבר לטופס הוספת פריט

{

AddItem newItem = new AddItem();

newItem.ShowDialog();

}

private void btnAddQuestion\_Click(object sender, EventArgs e) //מעבר לטופס בחירת סוג שאלה

{

ChooseQuestion chooseQuestion = new ChooseQuestion();

chooseQuestion.ShowDialog();

}

private void btnBack\_Click(object sender, EventArgs e) //סגירת הטופס וחזרה לתפריט הראשי

{

this.Close();

}

}

}

**:** **AddItem טופס**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

public partial class AddItem : Form //טופס הוספת פריט מידע

{

public AddItem()

{

InitializeComponent();

Globals.LoadInfoData(); //טעינת קובץ המידע מחדש ועדכון הרשימה

pbxAddPic.Visible = false;

}

private void btnBack\_Click(object sender, EventArgs e) //כפתור חזרה לתפריט הקודם

{

DialogResult dr = MessageBox.Show("Are you sure?","Addition didn't save",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dr == System.Windows.Forms.DialogResult.Yes)

{

this.Close();

Clear();

}

}

private void btnAddPic\_Click(object sender, EventArgs e) //כפתור העלאת תמונה מסייר קבצים

{

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

pbxAddPic.ImageLocation = ofd.FileName;

pbxAddPic.Visible = true;

}

}

private void btnApprove\_Click(object sender, EventArgs e) //בדיקת תקינות נתונים

{

try

{

if (txtAddTopic.Text == "" || txtAddContact.Text == "" ||

(pbxAddPic.ImageLocation == null && pbxAddPic.Visible == true)) //בדיקה האם כל השדות מלאים

{

MessageBox.Show("All fields must be filled!");

return;

}

foreach (var item in Globals.infoData) //בדיקה האם הפריט לא קיים כבר בקובץ

{

if (item.Content == txtAddContact.Text)

{

MessageBox.Show("The item is already exist");

return;

}

}

string rowNumber = (Globals.infoData.Count() + 1).ToString();

if (pbxAddPic.ImageLocation == null) //כתיבה לתוך הקובץ- פריט מידע ללא תמונה

{

string newRow = "\r\n" + rowNumber + ';' + txtAddTopic.Text + ';' + txtAddContact.Text;

File.AppendAllText("DATA\\infoData.txt", newRow);

}

else

{

//שמירת תמונה בשם הרצוי והעלאת תמונה לתקיית DIMAGES

string imageName = rowNumber + txtAddTopic.Text + pbxAddPic.ImageLocation.Substring(pbxAddPic.ImageLocation.LastIndexOf('.'));

File.Copy(pbxAddPic.ImageLocation, "DATA\\DIMAGES\\" + imageName);

string newRow = "\r\n" + rowNumber + ';' + txtAddTopic.Text + ';' + txtAddContact.Text + ';' + imageName;

File.AppendAllText("DATA\\infoData.txt", newRow);

}

Globals.LoadInfoData(); //טעינת קובץ המידע מחדש

MessageBox.Show("The item was added seccessfully");

this.Close();

Clear();

}

catch (FileNotFoundException)

{

MessageBox.Show("File Not Found");

}

catch (OutOfMemoryException)

{

MessageBox.Show("Insufficient memory to allocate a buffer for the returned string Exception");

}

catch (Exception m)

{

MessageBox.Show("Exception" + m.Message);

}

}

private void Clear() //מתודה שמאפסת את הפקדים למצב התחלתי

{

txtAddTopic.Text = "";

txtAddContact.Text = "";

pbxAddPic.ImageLocation = null;

pbxAddPic.Visible = false;

}

}

}

**:** **ChooseQuestion טופס**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

public partial class ChooseQuestion : Form //טופס בחירת סוג השאלה אותה המשתמש רוצה להוסיף

{

public static int type=0; //משתנה עזר שעוזר לדעת באיזה סוג שאלה מדובר

public ChooseQuestion()

{

InitializeComponent();

}

private void btnAddQuestion3\_Click(object sender, EventArgs e) //פתיחת הטופס המתאים עפ"י סוג השאלה

{

AddQuestionWImage MultipaleWImage = new AddQuestionWImage();

MultipaleWImage.ShowDialog();

}

private void btnBack\_Click(object sender, EventArgs e) //כפתור חזרה לטופס הקודם

{

this.Close();

}

private void btnAddQuestion2\_Click(object sender, EventArgs e) //פתיחת הטופס המתאים עפ"י סוג השאלה

{

type = 2;

AddQuestion TrueOrFalseWImage = new AddQuestion();

TrueOrFalseWImage.ShowDialog();

}

private void btnAddQuestion1\_Click(object sender, EventArgs e) //פתיחת הטופס המתאים עפ"י סוג השאלה

{

type = 1;

AddQuestion Multipale = new AddQuestion();

Multipale.ShowDialog();

}

private void btnAddQuestion0\_Click(object sender, EventArgs e) //פתיחת הטופס המתאים עפ"י סוג השאלה

{

type = 0;

AddQuestion TrueOrFalse = new AddQuestion();

TrueOrFalse.ShowDialog();

}

}

}

**:** **AddQuestion טופס**

using System;

using System.IO;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GeometricShapes

{

public partial class AddQuestion : Form //טופס הוספת שאלה

{

public AddQuestion()

{

InitializeComponent();

this.ClientSize = new System.Drawing.Size(740, 550);

Globals.LoadGameData(); //טעינת קובץ המידע של המשחק מחדש ועדכון המערך

Show();

}

private void btnBack\_Click(object sender, EventArgs e) //כפתור חזרה לתפריט הקודם

{

DialogResult dr = MessageBox.Show("Are you sure?", "Addition didn't save",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dr == System.Windows.Forms.DialogResult.Yes)

{

this.Close();

Clear();

}

}

private void Show() //מתודה שמציגה פקדים לפי סוג השאלה שאותה המשתמש רוצה להוסיף

{

if (ChooseQuestion.type == 1)

{

txtIncorrect2.Visible = true;

lblIncorrect2.Visible = true;

}

else if(ChooseQuestion.type == 2)

{

pcbQuestion.Visible = true;

btnAddPic.Visible = true;

}

return;

}

private void Clear() //מתודה שמאפסת את הפקדים

{

txtAddQuestion.Text = "";

txtCorrectAnswer.Text = "";

txtIncorrect1.Text = "";

txtIncorrect2.Text = "";

pcbQuestion.Image = null;

}

private void btnAddPic\_Click(object sender, EventArgs e) //כפתור העלאת תמונה מסייר קבצים

{

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

pcbQuestion.ImageLocation = ofd.FileName;

}

}

private void btnApprove\_Click(object sender, EventArgs e) //בדיקת תקינות נתונים

{

try

{

if (txtAddQuestion.Text == "" || txtCorrectAnswer.Text == "" || txtIncorrect1.Text == "") //בדיקה האם כל השדות מלאים

{

MessageBox.Show("All fields must be filled!");

return;

}

foreach (Questions item in Globals.gameData) //בדיקה האם השאלה לא קיימת כבר בקובץ

{

if (item.Question == txtAddQuestion.Text)

{

MessageBox.Show("The question is already exist");

return;

}

}

// כתיבה לתוך הקובץ עפ"י סוג שאלה ושמירת תמונת השאלה בתקיית QIMAGES

string rowNumber = (Globals.gameData.Count + 1).ToString();

string newRow;

if (txtIncorrect2.Text != "")

{

newRow = "\r\n" + rowNumber + ';' + (int)Qtype.Multiple + ';' + txtAddQuestion.Text + ';' + txtCorrectAnswer.Text + ';' + txtIncorrect1.Text + ';' + txtIncorrect2.Text;

}

else if (pcbQuestion.ImageLocation != null)

{

string image = rowNumber + "Question" + pcbQuestion.ImageLocation.Substring(pcbQuestion.ImageLocation.LastIndexOf('.'));

File.Copy(pcbQuestion.ImageLocation, "DATA\\QIMAGES\\" + image);

newRow = "\r\n" + rowNumber + ';' + (int)Qtype.TrueFalseWImage + ';' + txtAddQuestion.Text + ';' + txtCorrectAnswer.Text + ';' + txtIncorrect1.Text + ';' + image;

}

else

{

newRow = "\r\n" + rowNumber + ';' + (int)Qtype.TrueFalse + ';' + txtAddQuestion.Text + ';' + txtCorrectAnswer.Text + ';' + txtIncorrect1.Text;

}

File.AppendAllText("DATA\\gameData.txt", newRow);

Globals.LoadGameData(); //טעינת קובץ המידע של המשחק מחדש ועדכון המערך

MessageBox.Show("The question was added seccessfully");

this.Close();

Clear();

}

catch (FileNotFoundException)

{

MessageBox.Show("File Not Found");

}

catch (OutOfMemoryException)

{

MessageBox.Show("Insufficient memory to allocate a buffer for the returned string Exception");

}

catch (Exception m)

{

MessageBox.Show("Exception" + m.Message);

}

}

}

}

**:** **AddQuestionWImage טופס**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace GeometricShapes

{

public partial class AddQuestionWImage : Form //טופס הוספת שאלה עם תמונה

{

public AddQuestionWImage()

{

InitializeComponent();

this.ClientSize = new System.Drawing.Size(750, 600);

Globals.LoadGameData(); //טעינת קובץ המידע של המשחק מחדש ועדכון המערך

}

private void btnBack\_Click(object sender, EventArgs e) //כפתור חזרה לתפריט הקודם

{

DialogResult dr = MessageBox.Show("Are you sure?", "Addition didn't save",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dr == System.Windows.Forms.DialogResult.Yes)

{

this.Close();

Clear();

}

}

private void btnCorrectAnswer\_Click(object sender, EventArgs e) //כפתור העלאת תמונת תשובה נכונה מסייר קבצים

{

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

pcbCorrect.ImageLocation = ofd.FileName;

}

}

private void btnIncorrect1\_Click(object sender, EventArgs e) //כפתור העלאת תמונה שגויה 1 מסייר קבצים

{

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

pcbIncorrect1.ImageLocation = ofd.FileName;

}

}

private void btnIncorrect2\_Click(object sender, EventArgs e) //כפתור העלאת תמונה שגויה 2 מסייר קבצים

{

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

pcbIncorrect2.ImageLocation = ofd.FileName;

}

}

private void Clear() //מתודה שמאפסת את הפקדים

{

pcbIncorrect1.ImageLocation = null;

pcbIncorrect2.ImageLocation = null;

pcbCorrect.ImageLocation = null;

txtAddQuestion.Text = "";

}

private void btnApprove\_Click(object sender, EventArgs e) //בדיקת תקינות נתונים

{

try

{

if (txtAddQuestion.Text == "" || pcbCorrect.ImageLocation == null ||

pcbIncorrect2.ImageLocation == null || pcbIncorrect1.ImageLocation == null) //בדיקה האם כל השדות מלאים

{

MessageBox.Show("All fields must be filled!");

return;

}

foreach (Questions item in Globals.gameData) //בדיקה האם השאלה לא קיימת כבר בקובץ

{

if (item.Question == txtAddQuestion.Text)

{

MessageBox.Show("The question is already exist");

return;

}

}

// כתיבה לתוך הקובץ ושמירת תמונות בתקיית QIMAGES

string prefix = (Globals.gameData.Count + 1).ToString();

string image1 = prefix + "Correct" + pcbCorrect.ImageLocation.Substring(pcbCorrect.ImageLocation.LastIndexOf('.'));

string image2 = prefix + "Incorrect1" + pcbIncorrect1.ImageLocation.Substring(pcbIncorrect1.ImageLocation.LastIndexOf('.'));

string image3 = prefix + "Incorrect2" + pcbIncorrect2.ImageLocation.Substring(pcbIncorrect2.ImageLocation.LastIndexOf('.'));

File.Copy(pcbCorrect.ImageLocation, "DATA\\QIMAGES\\" + image1);

File.Copy(pcbIncorrect1.ImageLocation, "DATA\\QIMAGES\\" + image2);

File.Copy(pcbIncorrect2.ImageLocation, "DATA\\QIMAGES\\" + image3);

string newRow = "\r\n" + (Globals.gameData.Count + 1).ToString() + ';' + (int)Qtype.MultipleWImages + ';' + txtAddQuestion.Text + ';' + image1 + ';' + image2 + ';' + image3;

File.AppendAllText("DATA\\gameData.txt", newRow);

Globals.LoadGameData(); //טעינת קובץ המידע של המשחק מחדש ועדכון המערך

MessageBox.Show("The question was added seccessfully");

this.Close();

Clear();

}

catch (FileNotFoundException)

{

MessageBox.Show("File Not Found");

}

catch (OutOfMemoryException)

{

MessageBox.Show("Insufficient memory to allocate a buffer for the returned string Exception");

}

catch (Exception m)

{

MessageBox.Show("Exception" + m.Message);

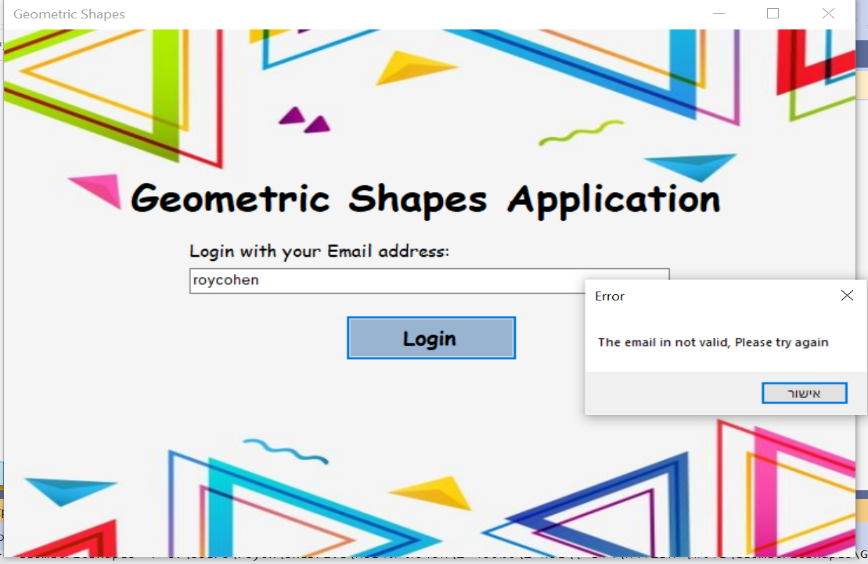
}

}

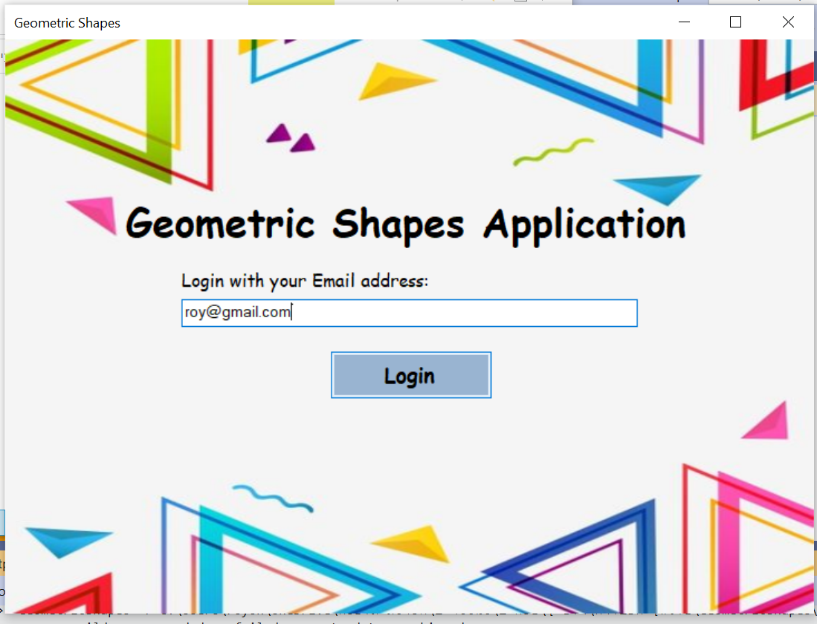
}

}

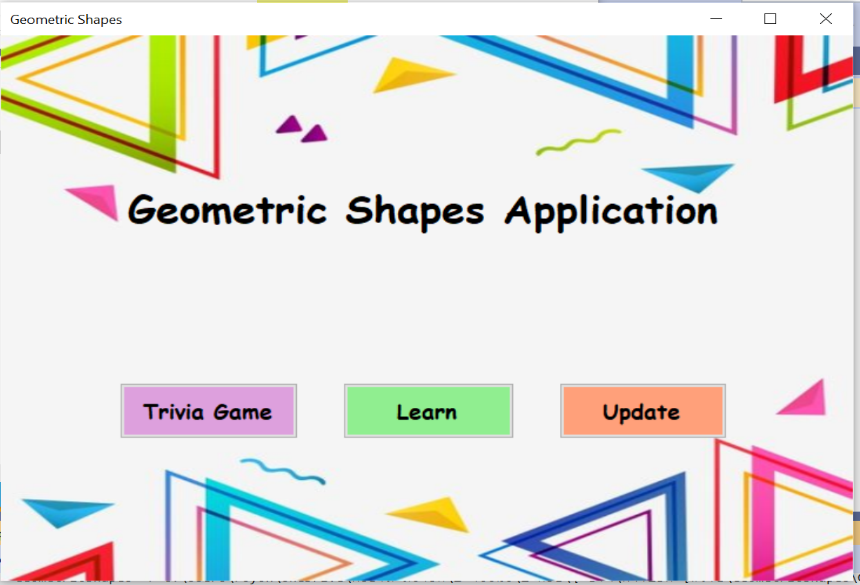
**פלטי המסך:**

****

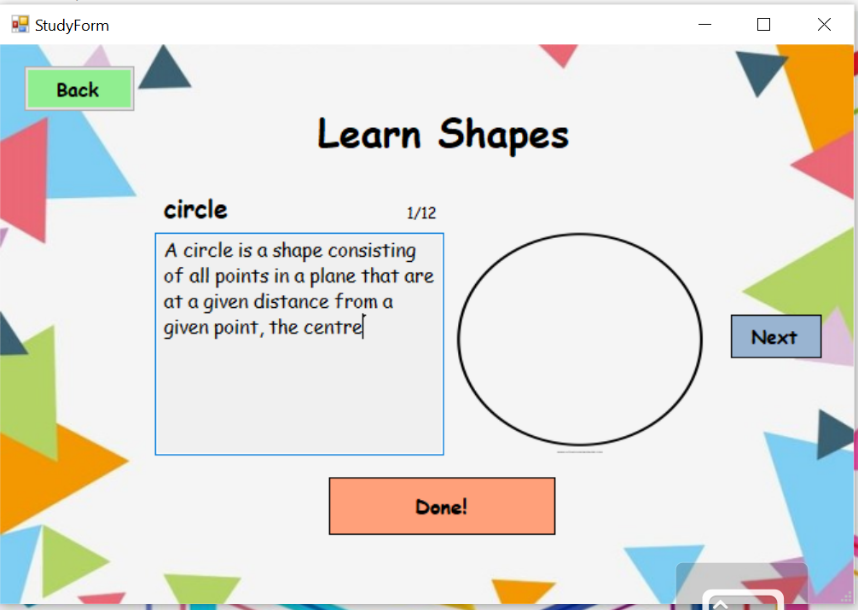
**שגיאה בהזנת כתובת mail**

****

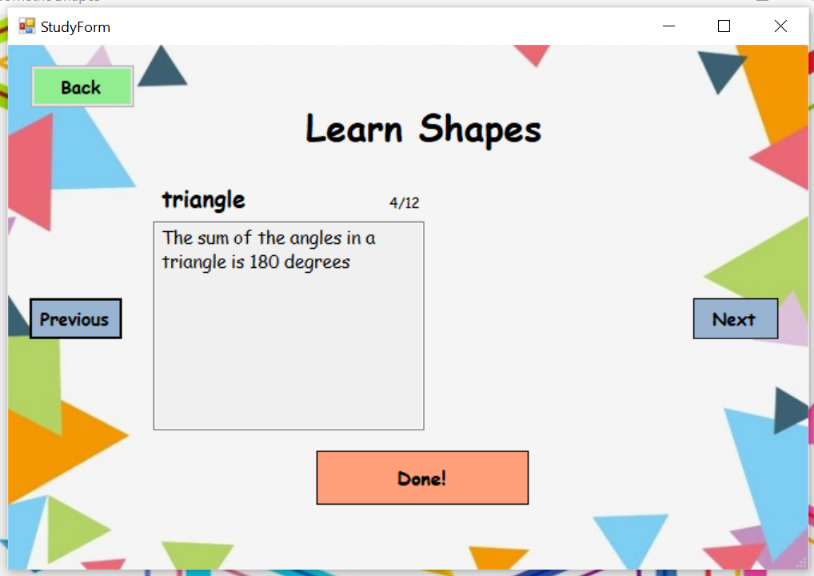
**כתיבת כתובת מייל נכונה**

****

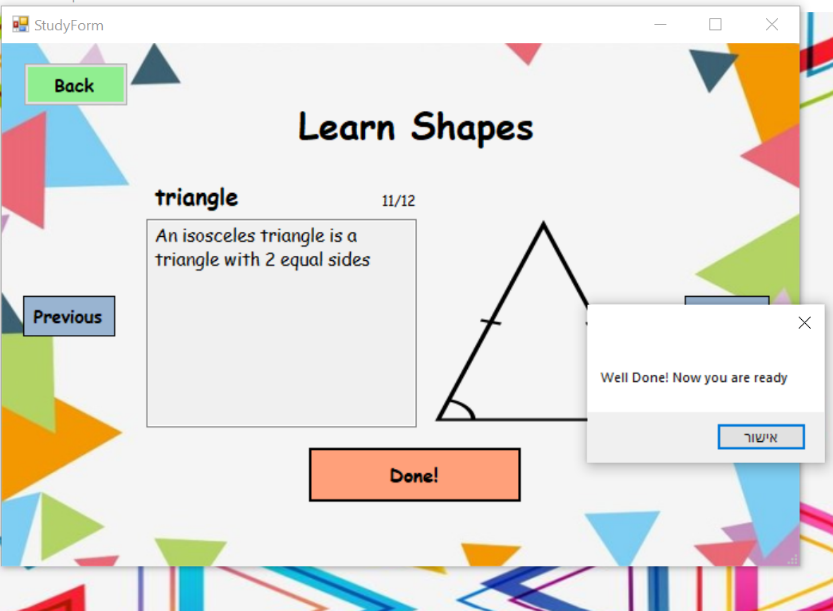
**מסך ראשי**

****

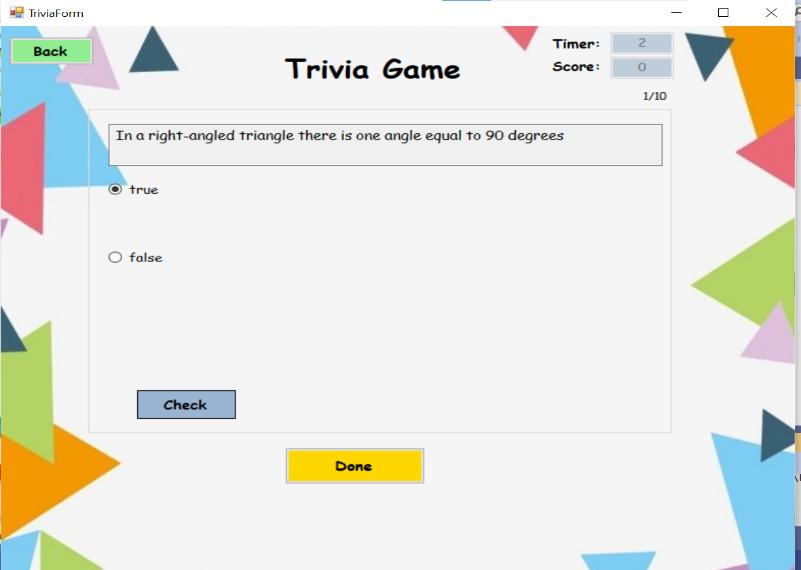
**מסך מידע עם תמונה**

****

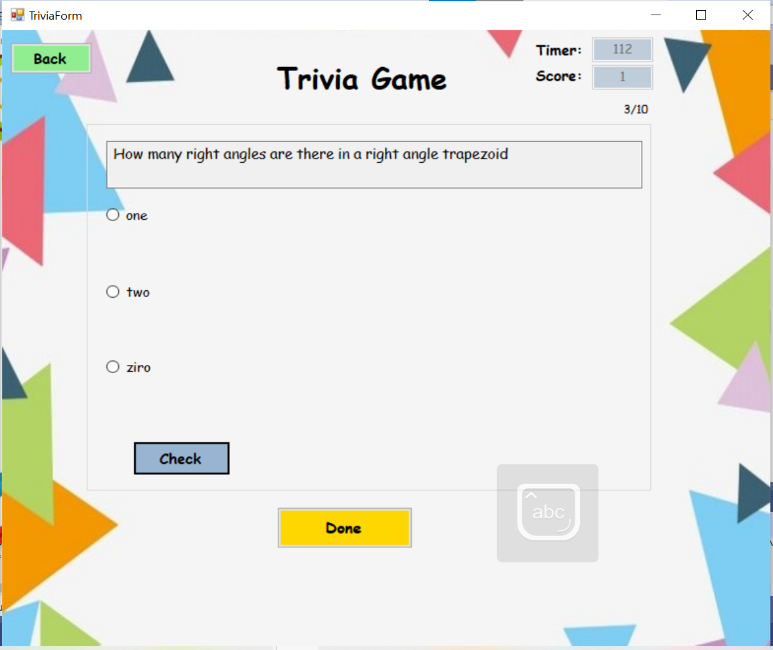
**מסך מידע בלי תמונה**

****

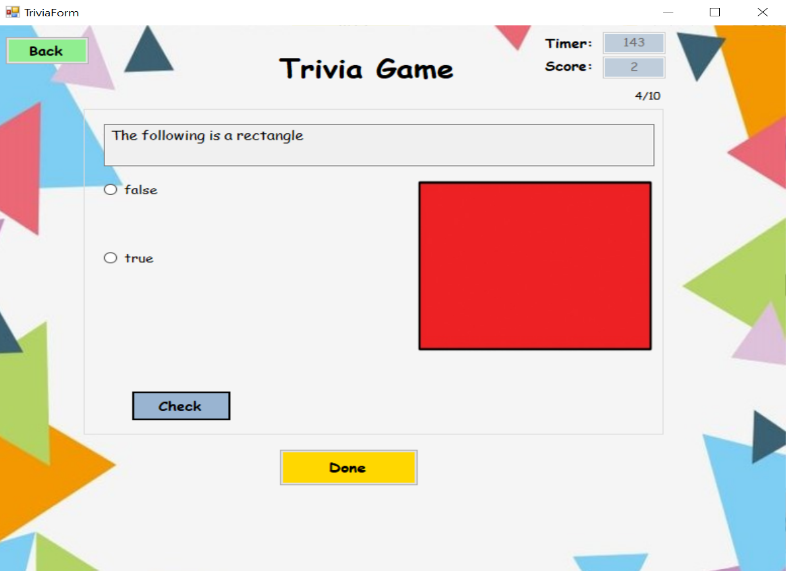
**סיום מסך מידע**

****

**שאלה מסוג נכון לא נכון**

****

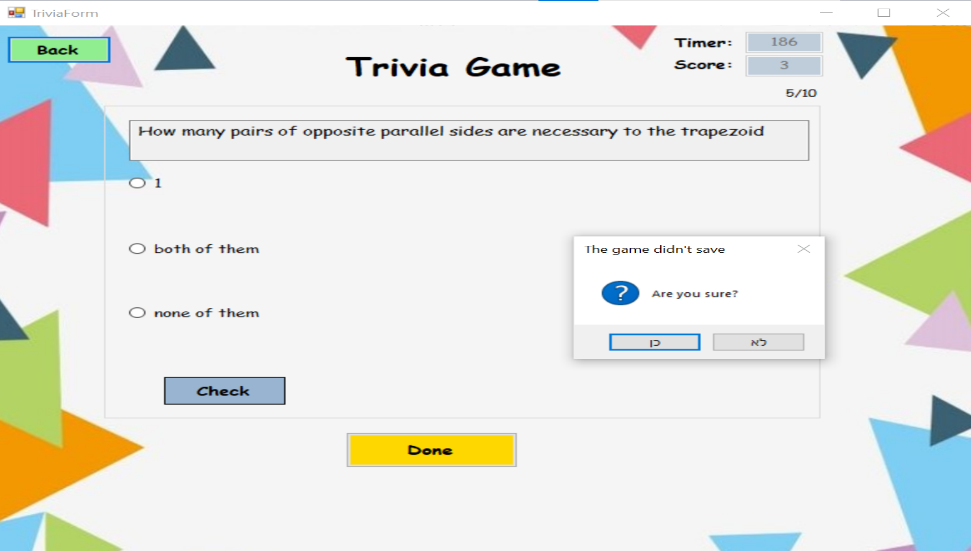
**שאלות רב בחירה**

****

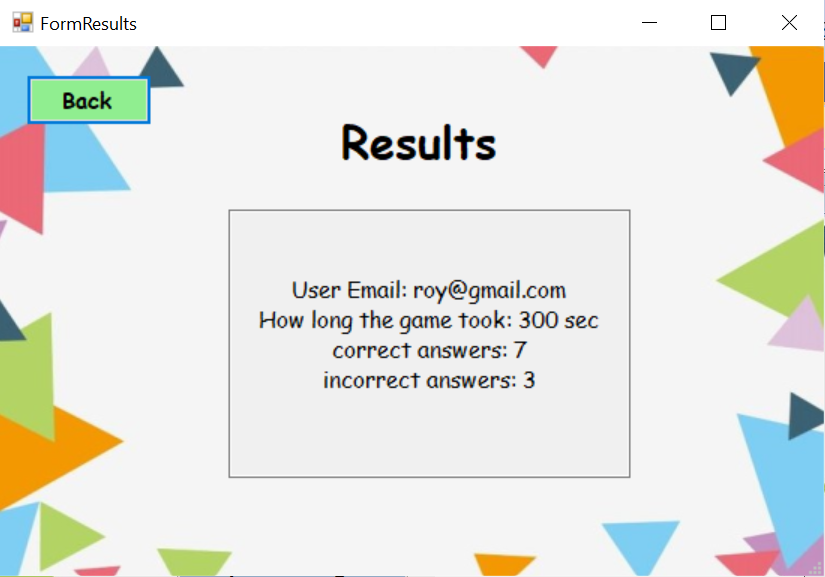
**נכון\ לא נכון עם תמונה**

****

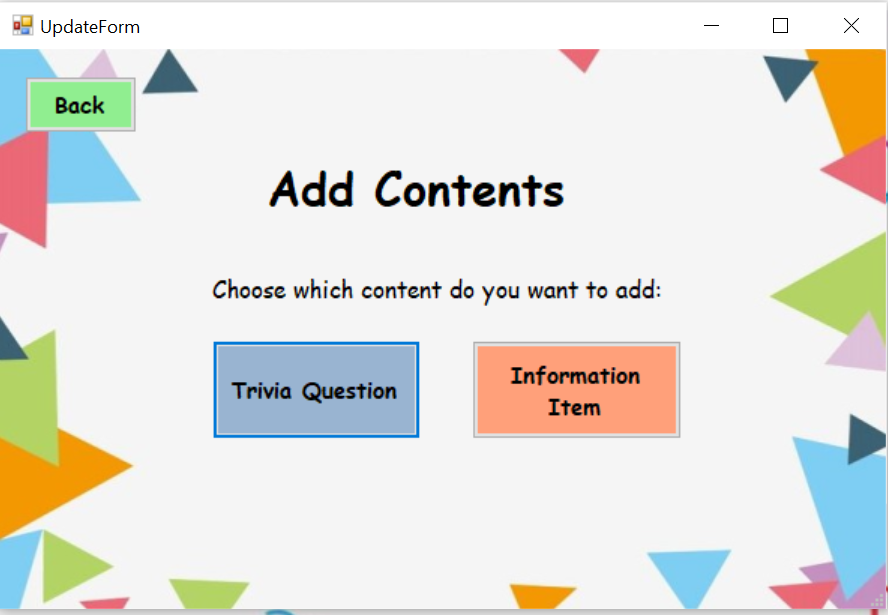
**רב בחירה שהתשובות הן תמונות**

****

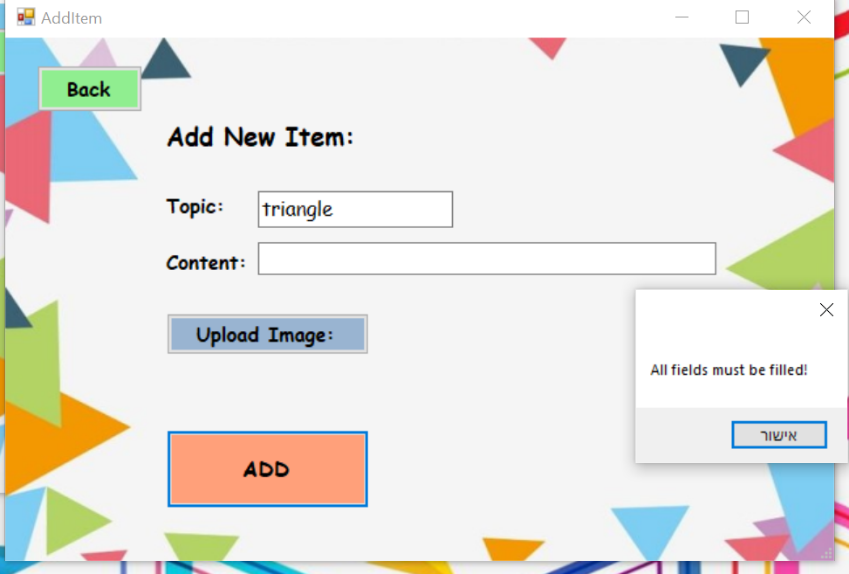
**לחיצה על פקד "Back" ואי שמירת המשחק**

****

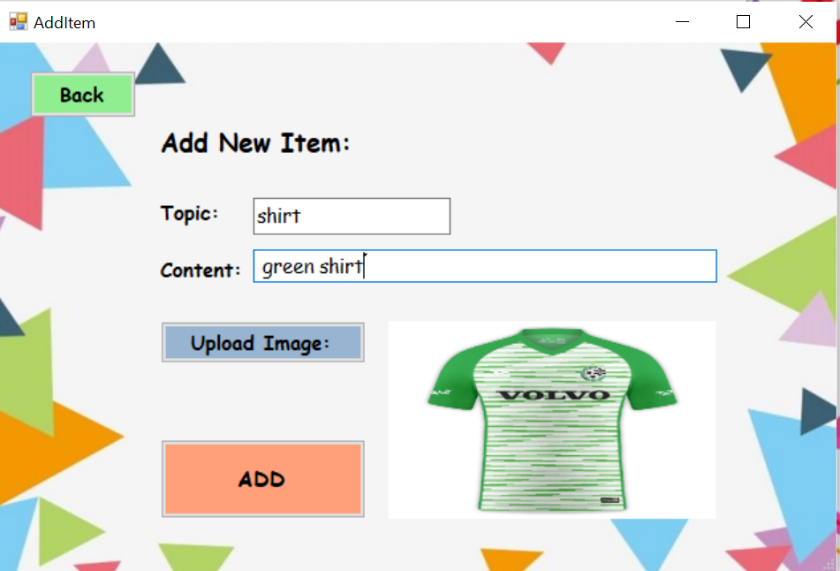
**דף הצגת התשובות לאחר סיום המשחק.**

****

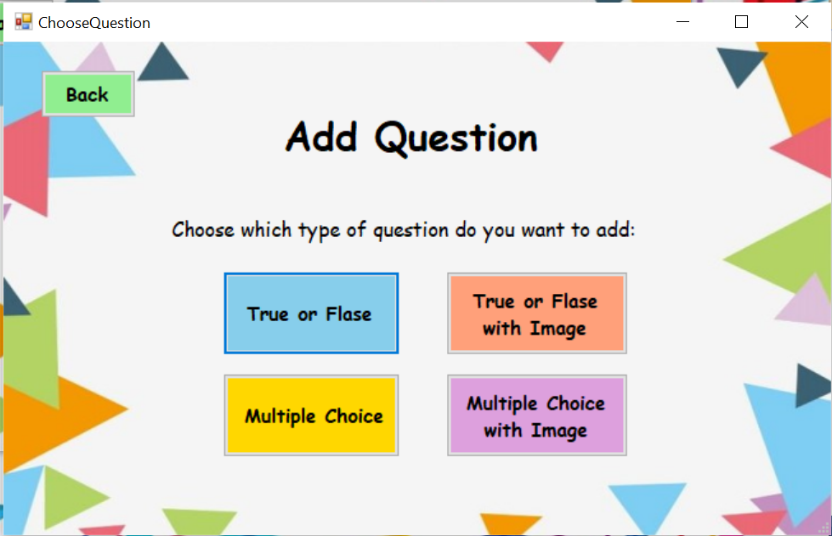
**תפריט הוספת תוכן**

****

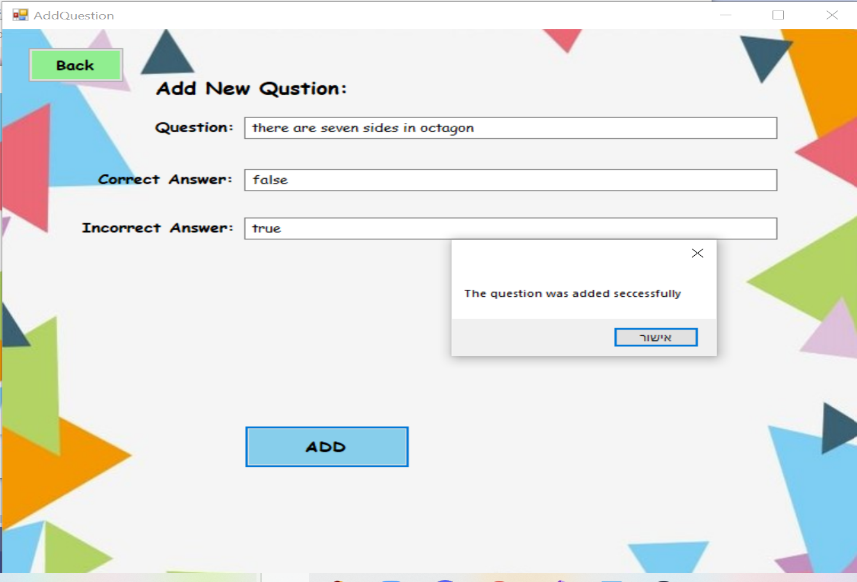
**שגיאה בהוספת פריט מידע**

****

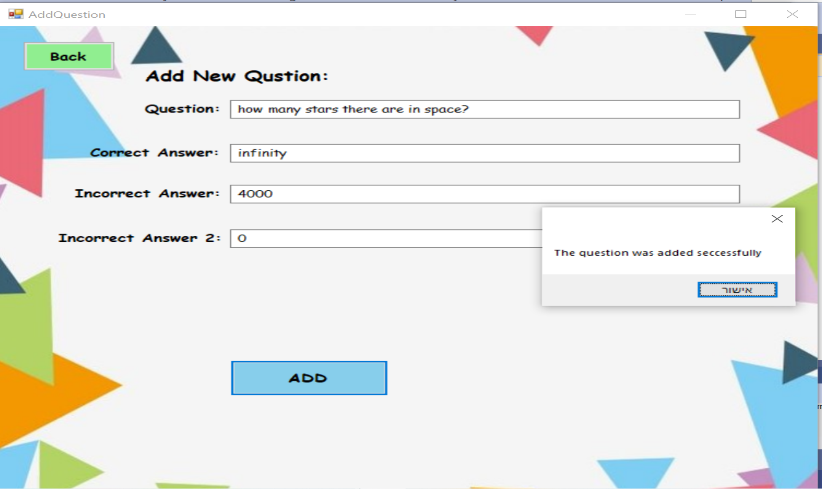
**הוספת פריט מידע עם תמונה**

****

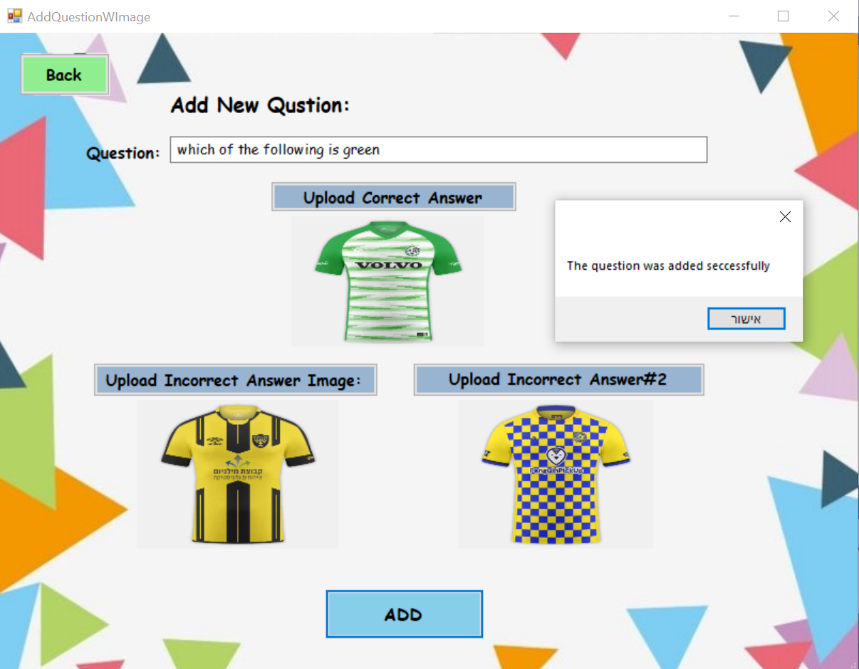
**תפריט הוספת שאלה**

****

**הוספת שאלה נכון/לא נכון**

****

**הוספת שאלת רב בחירה**

****

**רב בחירה, שהתשובות הן תמונות**

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

**gameDATA.txt עם השאלות החדשות**