

Colegiul Național “Unirea”

Proiect atestat profesional

Tema: aplicație pentru calculator

Autor: Neață Adrian

Profesor îndrumător: Colin Liliana

Cuprins

1. Prezentarea temei	pag 3
2. Obiectivele proiectului.....	pag 3
3. Sursele de informare.....	pag 3
4. Analiza proiectului.....	pag 3
5. Proiectarea.....	pag 4
6. Implementarea.....	pag 4
7. Resurse necesare prezentării lucrării.....	pag 10
8. Bibliografia.....	pag 10

1. Prezentarea temei

Am vrut să fac ceva original și aveam de mult ideea de a crea o aplicație pentru calculator. Din moment ce acesta este ultimul meu an de liceu am vrut să realizez un joc trivia care să-mi testeze cunoștințele de informatică pe care le-am acumulat de-a lungul anilor și astfel a luat ființă proiectul meu.

2. Obiectivele proiectului

Obiectivul proiectului meu este de a oferi o listă relativ lungă de întrebări cu câte patru răspunsuri din limbajul de programare C++ dintre care unul singur să fie corect. Proiectul nu trebuie să aibă un design complex; n-am oferit explicații despre modul de joacă deoarece nu am considerat necesară această precizare.

3. Sursele de informare

Singurela sursă de informare a fost site-ul de unde am selectat întrebările pe care le-am folosit pentru crearea proiectului.
(http://www.cnszalau.ro/wp-content/uploads/2014/03/2009_Variante_Info_rmatica.pdf)

4. Analiza proiectului

Aplicația este realizată în programul Unity, iar scripturile în MonoDevelop-Unity. Pentru crearea unei baze de date în care să fie memorate cele necesare proiectului am folosit DB Browser for SQLite.

Jocul are un meniu principal cu trei butoane: “Start” pentru a începe să răspunzi la întrebări, “Ajutor” pentru a oferi niște informații despre obiectivul jocului și “Ieșire” pentru a închide aplicația.

În momentul în care se apasă pe butonul “Start”, jucătorul este trimis într-o altă scenă unde este prezentă o întrebare, dedesubtul căreia se află patru butoane cu câte un răspuns fiecare și un buton “Înapoi” care întoarce jucătorul la meniul principal. Apăsarea unuia dintre răspunsuri va colora butonul respectiv în verde dacă răspunsul este corect și în roșu dacă este greșit. De asemenea va apărea după apăsare un alt buton “Următoarea” care va trimite jucătorul la următoarea întrebare.

5. Proiectarea

Baza de date conține două tabele: “InformaticaC” unde sunt puse întrebările și fiecareia i se dă câte un ID pentru a le distinge și “RaspunsuriInformaticaC” unde apar toate răspunsurile la întrebări. Aceste răspunsuri au un ID identic cu cel al întrebării de unde provine (Foreign Key), de asemenea mai au și o coloană “corect” cu o valoare nulă pentru întrebările greșite și 1 pentru cele corecte.

6. Implementarea

Pe de parte cea mai dificilă operație a fost crearea scripturilor deoarece a fost nevoie de folosirea codurilor SQL prin intermediul limbajului C#.

Pentru crearea scriptului care atribuie aleator fiecărui buton un răspuns a trebuit să iau un vector în care să memorez pozițiile butoanelor pe care deja le-am ocupat (pozițiile butoanelor sunt în ordine de la dreapta la stânga, de sus în jos 0, 1, 2, 3) și am creat o funcție recursivă care să continue să-mi aleagă unul din butoane la întâmplare să pună un răspuns până găsește acea poziție goală.

Verificarea răspunsului corect se face în felul următor: dacă textul butonului pe care a apăsă jucătorul este identic cu răspunsul din baza de date corect, corespunzător întrebării date, atunci butonul se colorează verde, altfel roșu.

```

1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4 using Mono.Data.Sqlite;
5
6 public class AssignQuestion : MonoBehaviour
7 {
8     private Text question;
9     public static int IndexC;
10    private SqlCommand command;
11    private SqlDataReader reader;
12
13    void Awake()
14    {
15        IndexC = Random.Range(0, 100);
16        question = GetComponent<Text>();
17        var connection = new SqlConnection("URI=file:" + Application.dataPath + "/StreamingAssets/Questions.db");
18        connection.Open();
19        command = connection.CreateCommand();
20        command.CommandText = "SELECT Intrebare FROM Informatiicac WHERE ID = " + IndexC;
21        question.text = command.ExecuteScalar().ToString();
22        connection.Close();
23    }
24
25
26 }

```

Scriptul folosit pentru afișarea întrebării

```

1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4 using Mono.Data.Sqlite;
5 public class AnswerShown : MonoBehaviour
6 {
7     Button button;
8     Text txt;
9     SqlCommand command;
10
11     void Start()
12     {
13         button = GetComponent<Button>();
14         txt = button.GetComponentInChildren<Text>();
15         var connection = new SqliteConnection("URI=file:" + Application.dataPath + "/StreamingAssets/Questions.db");
16         connection.Open();
17         command = connection.CreateCommand();
18         command.CommandText = "SELECT raspuns FROM RaspunsuriInformatiicac WHERE ID = " + AssignQuestion.Indexc + " AND correct = 1";
19         if (txt.text == command.ExecuteScalar().ToString())
20         {
21             button.image.color = Color.green;
22         }
23         else button.interactable = false;
24         connection.Close();
25     }
26 }

```

Scriptul folosit pentru colorarea butonului cu răspunsul corect cu verde

```

1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4 using Mono.Data.Sqlite;
5
6 public class CheckAnswer : MonoBehaviour {
7     Button button, buttoni;
8     Text txt, answer;
9     RectTransform panel;
10    int i;
11    SqliteCommand command;
12    // Use this for initialization
13    void Start () {
14        button = GetComponent<Button>();
15        txt = button.GetComponentInChildren<Text>();
16        var connection = new SqliteConnection("URI=file:" + Application.dataPath + "/StreamingAssets/Questions.db");
17        connection.Open();
18        command = connection.CreateCommand();
19        command.CommandText = "SELECT raspuns FROM RaspunsuriInformatiicac WHERE ID = " + AssignQuestion.Indexc + " AND corect = 1";
20        if (txt.text == command.ExecuteScalar().ToString())
21        {
22            button.image.color = Color.green;
23        }
24        else
25        {
26            button.image.color = Color.red;
27            button.interactable = false;
28        }
29        connection.Close();
30    }
31 }

```

Scriptul care verifică dacă răspunsul ales de jucător este corect colorându-l verde

```

1 using UnityEngine;
2 using UnityEngine.UI;
3 using System.Collections;
4 using Mono.Data.Sqlite;
5
6 public class AssignAnswer : MonoBehaviour
7 {
8     private int[] exception = new int[] { -1, -1, -1, -1 };
9     Text answer;
10    private Button button;
11    private RectTransform panel;
12    int i = 0, x, y;
13    SQLiteCommand command;
14    int Index()
15    {
16        int c = 1, j;
17        x = Random.Range(0, 4);
18        for (j = 0; j < i; j++)
19            if (exception[j] == x) c = 0;
20        if (c == 1)
21        {
22            exception[i] = x;
23            i++;
24            return x;
25        }
26        else return (Index());
27    }
28    // creates a random value then checks if it hasn't been before
29
30    void Start()
31    {
32        var connection = new SQLiteConnection("URI=file:" + Application.dataPath + "/StreamingAssets/Questions.db");
33        panel = GetComponent<RectTransform>();
34        connection.Open();
35        command = connection.CreateCommand();
36        command.CommandText = "SELECT raspuns FROM RaspunsuriInformaticaC WHERE ID = " + AssignQuestion.IndexC;
37        SQLiteDataReader reader = command.ExecuteReader();
38        while (reader.Read())
39        {
40            GameObject button = panel.transform.GetChild(Index()).gameObject;
41            answer = button.GetComponentInChildren<Text>();
42            answer.text = reader["raspuns"].ToString();
43        }
44        connection.Close();
45        // gives each button a different answer
46    }
47 }

```

Scriptul care atribuie aleator fiecărui buton un răspuns


```

1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.SceneManagement;
4
5 public class LoadSceneOnClick : MonoBehaviour {
6
7     public void LoadByIndex(int sceneIndex)
8     {
9         SceneManager.LoadScene(sceneIndex);
10    }
11 }

```

Scriptul care încarcă scena

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class QuitOnClick : MonoBehaviour {
5
6     public void Quit ()
7     {
8         #if UNITY_EDITOR
9             UnityEditor.EditorApplication.isPlaying = false;
10        #else
11            Application.Quit ();
12        #endif
13    }
14 }

```

Scriptul care încheie jocul

7. Resurse necesare prezentării lucrării

A. Resurse hard:

- un calculator

B. Resurse software:

- orice versiune superioară de Windows98

8. Bibliografie

- forumul site-ului Unity <https://forum.unity3d.com/> ;
- <https://www.w3schools.com/sql/DEfaULT.asP> ;
- <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx> ;
- <https://docs.unity3d.com/Manual/index.html> ;
- http://www.cnszalau.ro/wp-content/uploads/2014/03/2009_Variante_Informatica.pdf ;
- https://www.youtube.com/watch?v=IaspFwXGprg&list=PLX-uZVK_0K_7NmsYfe2BTOk_lamWC2kU3 ;