

## Sumari

<b>SUMARI</b>	<hr/> 1
<b>ANNEX A: FITXER XML</b>	<hr/> 3
<b>ANNEX B: SCRIPTS</b>	<hr/> 5
AspectUtility.....	5
Boto_joc .....	7
Boto_opcions1 .....	7
Boto_resultats .....	8
Boto_sortir .....	8
Canvi_color .....	8
Opcions .....	9
Botonumero.....	11
Altaveu .....	12
Joc.....	12
Comparador .....	16
Crear_txt.....	20
Resultats .....	21



## Annex A: Fitxer XML

```
<?xml version="1.0" encoding="utf-8"?>
<JOC>
  <dadesCAT>
    <imatge0 nom="pilota" foto="pilota" so="pilotaSo" silabes="3" nivell="2">
      </imatge0>
    </dadesCAT>
    <dadesCAST>
      <imatge0 nom="pelota" foto="pilota" so="pelotaSo" silabes="3" nivell="2">
        </imatge0>
      </dadesCAST>
    </JOC>
```



## Annex B: Scripts

### AspectUtility

```
using UnityEngine;

public class AspectUtility : MonoBehaviour {

    public float _wantedAspectRatio = 1.77778f;
    static float wantedAspectRatio;
    static Camera cam;
    static Camera backgroundCam;
    void Awake()
    {
        cam = GetComponent<Camera>();
        if (!cam)
        {
            cam = Camera.main;
        }
        if (!cam)
        {
            Debug.LogError("No camera available");
            return;
        }
        wantedAspectRatio = _wantedAspectRatio;
        SetCamera();
    }
    public static void SetCamera()
    {
        float currentAspectRatio = (float)Screen.width / Screen.height;
        // If the current aspect ratio is already approximately equal to the desired
        aspect ratio,
        // use a full-screen Rect (in case it was set to something else previously)
        if ((int)(currentAspectRatio * 100) / 100.0f == (int)(wantedAspectRatio *
100) / 100.0f)
        {
            cam.rect = new Rect(0.0f, 0.0f, 1.0f, 1.0f);
            if (backgroundCam)
            {
                Destroy(backgroundCam.gameObject);
            }
            return;
        }
        // Pillarbox
        if (currentAspectRatio > wantedAspectRatio)
        {
            float inset = 1.0f - wantedAspectRatio / currentAspectRatio;
            cam.rect = new Rect(inset / 2, 0.0f, 1.0f - inset, 1.0f);
        }
        // Letterbox
        else
        {
            float inset = 1.0f - currentAspectRatio / wantedAspectRatio;
            cam.rect = new Rect(0.0f, inset / 2, 1.0f, 1.0f - inset);
        }
    }
}
```

```
if (!backgroundCam)
{
    // Make a new camera behind the normal camera which displays black;
otherwise the unused space is undefined
    backgroundCam = new GameObject("BackgroundCam",
typeof(Camera)).GetComponent<Camera>();
    backgroundCam.depth = int.MinValue;
    backgroundCam.clearFlags = CameraClearFlags.SolidColor;
    backgroundCam.backgroundColor = Color.black;
    backgroundCam.cullingMask = 0;
}
public static int screenHeight
{
    get
    {
        return (int)(Screen.height * cam.rect.height);
    }
}
public static int screenWidth
{
    get
    {
        return (int)(Screen.width * cam.rect.width);
    }
}
public static int xOffset
{
    get
    {
        return (int)(Screen.width * cam.rect.x);
    }
}
public static int yOffset
{
    get
    {
        return (int)(Screen.height * cam.rect.y);
    }
}
public static Rect screenRect
{
    get
    {
        return new Rect(cam.rect.x * Screen.width, cam.rect.y * Screen.height,
cam.rect.width * Screen.width, cam.rect.height * Screen.height);
    }
}
public static Vector3 mousePosition
{
    get
    {
        Vector3 mousePos = Input.mousePosition;
        mousePos.y -= (int)(cam.rect.y * Screen.height);
        mousePos.x -= (int)(cam.rect.x * Screen.width);
        return mousePos;
    }
}
public static Vector2 guiMousePosition
```

```

    {
        get
        {
            Vector2 mousePos = Event.current.mousePosition;
            mousePos.y = Mathf.Clamp(mousePos.y, cam.rect.y * Screen.height,
            cam.rect.y * Screen.height + cam.rect.height * Screen.height);
            mousePos.x = Mathf.Clamp(mousePos.x, cam.rect.x * Screen.width,
            cam.rect.x * Screen.width + cam.rect.width * Screen.width);
            return mousePos;
        }
    }
}

```

## Boto\_joc

```

using UnityEngine;

public class Boto_joc : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    void OnMouseDown()
    {
        Application.LoadLevel("pantalla_joc");
    }
}

```

## Boto\_opcions1

```

using UnityEngine;

public class Boto_opcions1 : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame

    void OnMouseDown()
    {
        Application.LoadLevel("pantalla_opcions");
    }
}

```

## Boto\_resultats

```
using UnityEngine;

public class Boto_resultats : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    void OnMouseDown()
    {
        Application.LoadLevel("pantalla_resultats");
    }
}
```

## Boto\_sortir

```
using UnityEngine;

public class Boto_sortir : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    void OnMouseDown()
    {
        Application.Quit();
    }
}
```

## Canvi\_color

```
using UnityEngine;

public class canvi_color : MonoBehaviour {

    private TextMesh t;

    // Use this for initialization
    void Start () {

    }
}
```



```

    void OnMouseEnter () {
        t = transform.GetComponent<TextMesh>();
        t.color = Color.yellow;
    }

    void OnMouseExit() {
        t = transform.GetComponent<TextMesh>();
        t.color = Color.white;
    }
}

```

## Opcions

```

using UnityEngine;
using UnityEngine.UI;

public class opciones : MonoBehaviour
{
    public static string idioma = "cat";
    public static int nivell = 2;
    private Image d, e, f, g, h;
    float amplada = (float)(Screen.width);
    float alcada = (float)(Screen.height);

    void Start()
    {
        seleccio_idi(idioma);
        seleccio_niv(nivell);

        GameObject.Find("Boto_baix").transform.position = new Vector3(amplada * 0.4f, alcada * 0.57f, 0);
        GameObject.Find("Boto_mig").transform.position = new Vector3(amplada * 0.55f, alcada * 0.57f, 0);
        GameObject.Find("Boto_alt").transform.position = new Vector3(amplada * 0.7f, alcada * 0.57f, 0);
        GameObject.Find("Boto_cat").transform.position = new Vector3(amplada * 0.45f, alcada * 0.35f, 0);
        GameObject.Find("Boto_cast").transform.position = new Vector3(amplada * 0.65f, alcada * 0.35f, 0);
        GameObject.Find("Boto_torna").transform.position = new Vector3(amplada * 0.75f, alcada * 0.15f, 0);
        GameObject.Find("Canvas").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada, alcada);
        GameObject.Find("Canvas2").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada, alcada);
        GameObject.Find("Boto_baix").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada * 0.12f, alcada * 0.12f);
        GameObject.Find("Boto_mig").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada * 0.12f, alcada * 0.12f);
        GameObject.Find("Boto_alt").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada * 0.12f, alcada * 0.12f);
        GameObject.Find("Boto_cat").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada * 0.15f, alcada * 0.12f);
        GameObject.Find("Boto_cast").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada * 0.15f, alcada * 0.12f);
    }
}

```

```
    GameObject.Find("Boto_torna").GetComponent<RectTransform>().sizeDelta = new  
Vector2(AMPLADA * 0.1f, ALCADA * 0.22f);  
  
}  
  
public void set_nivell(int x)  
{  
    nivell = x;  
    seleccio_niv(x);  
}  
public int nivell_value()  
{  
    return nivell;  
}  
  
public string idioma_value()  
{  
    return idioma;  
}  
  
public void set_idioma(string y)  
{  
    idioma = y;  
    seleccio_idi(y);  
}  
  
public void torna()  
{  
    Application.LoadLevel("pantalla_principal");  
}  
void seleccio_niv(int x)  
{  
    d = GameObject.Find("Boto_baix").GetComponent<Image>();  
    d.color = Color.white;  
    e = GameObject.Find("Boto_mig").GetComponent<Image>();  
    e.color = Color.white;  
    f = GameObject.Find("Boto_alt").GetComponent<Image>();  
    f.color = Color.white;  
  
    if (x == 1)  
    {  
        d = GameObject.Find("Boto_baix").GetComponent<Image>();  
        d.color = Color.yellow;  
    }  
    else if (x == 2)  
    {  
        d = GameObject.Find("Boto_mig").GetComponent<Image>();  
        d.color = Color.yellow;  
    }  
    else if (x == 3)  
    {  
        d = GameObject.Find("Boto_alt").GetComponent<Image>();  
        d.color = Color.yellow;  
    }  
}  
  
void seleccio_idi(string y)  
{
```

```

    if (y == "cat")
    {
        g = GameObject.Find("Boto_cat").GetComponent<Image>();
        g.color = Color.yellow;
        h = GameObject.Find("Boto_cast").GetComponent<Image>();
        h.color = Color.white;
    }
    else if (y == "cast")
    {
        h = GameObject.Find("Boto_cast").GetComponent<Image>();
        h.color = Color.yellow;
        g = GameObject.Find("Boto_cat").GetComponent<Image>();
        g.color = Color.white;
    }
}
}

```

## Botonumero

```

using UnityEngine;
using UnityEngine.UI;

public class botonumero : MonoBehaviour
{
    private Image correcte, incorrecte;
    public int x, a, b, c, d, e;
    private int silabes;
    private TextMesh t;
    private int g=0;

    void Start()
    {

    }

    void OnMouseEnter()
    {
        t = transform.GetComponent<TextMesh>();
        t.color = Color.yellow;
    }
    void OnMouseExit()
    {
        //Debug.Log("posa blanc " +g);
        if (g ==0)
        {
            //Debug.Log("entra if " + g);
            t = transform.GetComponent<TextMesh>();
            t.color = Color.white;
        }
        g = 0;
    }
    void OnMouseDown()
    {
        g = GameObject.Find("objectebuit2").GetComponent<joc>().clickat();
    }
}

```

```

//Debug.Log(g);
a = GameObject.Find("objectebuit2").GetComponent<joc>().get_apretat();
if (a==1)
{
    t = transform.GetComponent<TextMesh>();
    t.color = Color.yellow;
    //Debug.Log(t.color);
    silabes = GameObject.Find("objectebuit2").GetComponent<joc>().getsil();

    if (x == silabes)
    {
        correcte =
GameObject.Find("Correcte/Incorrecte").GetComponent<Image>();
        correcte.sprite = Resources.Load<Sprite>("tick");
        b =
GameObject.Find("objectebuit2").GetComponent<joc>().get_encertades();
        c=
GameObject.Find("objectebuit2").GetComponent<joc>().get_puntuacio();
    }
    else
    {
        correcte =
GameObject.Find("Correcte/Incorrecte").GetComponent<Image>();
        correcte.sprite = Resources.Load<Sprite>("wrong");
        e=
GameObject.Find("objectebuit2").GetComponent<joc>().get_fallades();
        d =
GameObject.Find("objectebuit2").GetComponent<joc>().get_puntuacio();

    }
}
}
}

```

## Altaveu

```

using UnityEngine;

public class altaveu : MonoBehaviour
{
    // Use this for initialization

    public void Reproduir()
    {
        GetComponent< AudioSource >().Play();
    }
}

```

## Joc

```

using UnityEngine;
using System.Collections.Generic;
using System.Xml;
using UnityEngine.UI;

```

```

using System;

public struct pack
{
    public string paraula;
    public Sprite foto;
    public AudioClip so;
    public int silabes;
    public pack(string n, string f, string s, string sil)
    {
        paraula = n;
        foto = Resources.Load<Sprite>(f);
        so = Resources.Load(s) as AudioClip;
        silabes = int.Parse(sil);
    }
}
public class joc : MonoBehaviour
{
    public pack[] paquet;
    private Image im, correcte, incorrecte;
    public AudioSource sound;
    private AudioClip sonido;
    public TextMesh nom;
    private int i,j,k,m,q=0,a=0,c=0,num;
    public List<int> total=new List<int>();
    public GameObject fletxa, s;
    public static int encertades = 0, fallades = 0;
    private TextMesh e, o, u, w, y,d;
    private int h = 0;

    float amplada = (float)(Screen.width);
    float alcada = (float)(Screen.height);

    // Use this for initialization
    void Main()
    {
        XmlDocument doc = new XmlDocument();
        doc.Load(@"C:\Users\itziar\Documents\universitat\quatri
9\TFG\unity\Assets\fitxerXML.xml");
        XmlNode taula;
        if (String.Equals(opcions.idioma, "cat"))
        {
            taula = doc.SelectSingleNode("/JOC/dadesCAT");
            d = GameObject.Find("Pregunta").GetComponent<TextMesh>();
            d.text = "Quin número de síl·labes té la paraula?";
        }
        else
        {
            taula = doc.SelectSingleNode("/JOC/dadesCAST");
            d = GameObject.Find("Pregunta").GetComponent<TextMesh>();
            d.text = "¿Qué número de silabas tiene la palabra?";
        }
        paquet = new pack[taula.ChildNodes.Count];

        for (int i=0; i<22; i++)
        //for (int i = 0; i < taula.ChildNodes.Count; i++)
        {
    
```

```

XmlAttribute nivell = taula.ChildNodes[i].Attributes["nivell"];
//Debug.Log("començà for "+i);

if (int.Parse(nivell.Value)<= opciones.nivell)
{
    //Debug.Log("entra if " + i);
    XmlAttribute nom = taula.ChildNodes[i].Attributes["nom"];
    //Debug.Log(nom);
    XmlAttribute foto = taula.ChildNodes[i].Attributes["foto"];
    //Debug.Log(foto);
    XmlAttribute so = taula.ChildNodes[i].Attributes["so"];
    //Debug.Log(so);
    XmlAttribute silabes = taula.ChildNodes[i].Attributes["silabes"];
    //Debug.Log(silabes);
    pack_elementpaquet = new pack(nom.Value, foto.Value, so.Value,
silabes.Value);
    paquet[h] = elementpaquet;
    //Debug.Log("suma llista possibles imatges " + h);
    h++;
}

}
doc.Save(@"C:\Users\itziar\Documents\universitat\quatri
9\TFG\unity\Assets\fitxerXML.xml");
}

void Start()
{
    GameObject.Find("Panel").transform.position = new Vector3(amplada/2,
alcada*0.55f, 0);
    GameObject.Find("Altaveu").transform.position = new Vector3(amplada*0.77f,
alcada / 2, 0);
    GameObject.Find("Següent").transform.position = new Vector3(amplada*0.765f,
alcada*0.15f, 0);
    GameObject.Find("Correcte/Incorrecte").transform.position = new
Vector3(amplada*0.65f, alcada / 2, 0);
    GameObject.Find("Micro").transform.position = new Vector3(amplada*0.3f,
alcada/2, 0);
    GameObject.Find("Canvas-Imatge").GetComponent<RectTransform>().sizeDelta =
new Vector2(amplada, alcada);
    GameObject.Find("Canvas-tick i
creu").GetComponent<RectTransform>().sizeDelta = new Vector2(amplada, alcada);
    GameObject.Find("Panel").GetComponent<RectTransform>().sizeDelta = new
Vector2(amplada / 5, alcada / 2.3f);
    GameObject.Find("Altaveu").GetComponent<RectTransform>().sizeDelta = new
Vector2(amplada / 8.7f, alcada / 3.7f);
    GameObject.Find("Següent").GetComponent<RectTransform>().sizeDelta = new
Vector2(amplada / 12f, alcada / 5f);

GameObject.Find("Correcte/Incorrecte").GetComponent<RectTransform>().sizeDelta = new
Vector2(amplada / 10.5f, alcada / 4.5f);
    GameObject.Find("Micro").GetComponent<RectTransform>().sizeDelta = new
Vector2(amplada / 8.5f, alcada / 3.5f);

for (int p = 0; p < 8; p++)
{
    k = UnityEngine.Random.Range(0, h);
    for (int s = 0; s <= p; s++)
    {

```

```

        if (s == p)
        {
            total.Add(k);
        }
    }

    j = total[q];
    visualitza(j);
    //Debug.Log("visualitza " + total[0]);

}

public void canvia()
{
    c = 0;
    a = 0;
    q++;
    //comparador.time = 0.0f;
    e = GameObject.Find("1").GetComponent<TextMesh>();
    e.color = Color.white;
    o = GameObject.Find("2").GetComponent<TextMesh>();
    o.color = Color.white;
    u = GameObject.Find("3").GetComponent<TextMesh>();
    u.color = Color.white;
    w = GameObject.Find("4").GetComponent<TextMesh>();
    w.color = Color.white;
    y = GameObject.Find("5").GetComponent<TextMesh>();
    y.color = Color.white;
    im = GameObject.Find("Micro").GetComponent<Image>();
    im.sprite = Resources.Load<Sprite>("micro");

    if (q < total.Count)
    {
        j = total[q];
        visualitza(j);
        correcte = GameObject.Find("Correcte/Incorrecte").GetComponent<Image>();
        correcte.sprite = Resources.Load<Sprite>("imatgeenblanc");
    }
    else
        Application.LoadLevel("pantalla_resultats");
}

void visualitza(int j)
{
    fletxa = GameObject.Find("Seguent");
    fletxa.SetActive(false);
    im = GameObject.Find("Imatge").GetComponent<Image>();
    im.sprite = paquet[j].foto;
    //Debug.Log(im);
    sound = GameObject.Find("objectebuit1").GetComponent< AudioSource >();
    //Debug.Log(sound);
    sound.clip= paquet[j].so;
    //Debug.Log(sound.clip);
    nom = GameObject.Find("Nom").GetComponent<TextMesh>();
    nom.text = paquet[j].paraula;
    //Debug.Log(nom.text);
}

```

```

}

public int getSil()
{
    fletxa.SetActive(true);
    //Debug.Log("getsil");
    return paquet[j].silabes;
}

public int get_apretat()
{
    a = a+1;
    return a;
}

public int get_encertades()
{
    encertades++;
    return encertades;
}

public int get_fallades()
{
    fallades++;
    return fallades;
}

public int get_puntuacio()
{
    return encertades*5-fallades;
}

public int clickat()
{
    c++;
    return c;
}

}

```

## Comparador

```

using UnityEngine;
using System.Collections.Generic;
using System;
using System.IO;
using UnityEngine.UI;

public class comparador : MonoBehaviour {

    private FFTWindow fftwindow;
    private AudioSource b;
    private float[] samples = new float[8192];
    private List<float> freq = new List<float>();

```

```

private List<float> f2 = new List<float>();
private int z, a = 0, r = 0, para = 0, sil = 0;
static float time = 0.0f;
private float m;
public Image im;
private GameObject fletxa;

// Use this for initialization
void Start()
{
}

void OnMouseDown()
{
    b = GameObject.Find("Micro").GetComponent();
    b.clip = Microphone.Start(null, false, 3, 44100); //Enregistra el so del
    microfon (el que hi hagi per defecte, per això el 0).           //El loop està posat a
    false perque nomes enregistri una vegada i no torni a començar el registre.
    //El 3 indica el temps
    de registre de l'audio.                                         //El 44100 la freqüència
    (velocitat de mostreig)

    while (!(Microphone.GetPosition(null) > 0))
    {
        b.Play();
    }      //Espera fins que comença el registre del microfon
    m = time;
    a = 1;
}

// Update is called once per frame
void Update()
{
    List<float> f1 = new List<float>();
    time = Time.time;
    if (a == 1)
    {
        if (String.Equals((time - m).ToString("f0"), "1") && para<=3)
        {
            b.GetSpectrumData(samples, 0, fftwindow);
            var audio_micro = File.CreateText("audio_micro.txt");
            for (int j = 0; j < samples.Length; j++)
            {
                f1.Add(1 / samples[j]);
                audio_micro.WriteLine(f1[j].ToString());
            }
            f2=guarda(f1);
            audio_micro.Close();
            para++;
        }
        if (String.Equals((time - m).ToString("f0"), "2") && r == 0)
        {
            b.Stop();
            r = 1;
            z = Tria(f2);
            sil = GameObject.Find("objectebuit2").GetComponent<joc>().getSil();
        }
    }
}

```

```

fletxa = GameObject.Find("Següent");
fletxa.SetActive(false);
if (z == sil)
{
    im = GameObject.Find("Micro").GetComponent<Image>();
    im.sprite = Resources.Load<Sprite>("smiley");
}
else
{
    im = GameObject.Find("Micro").GetComponent<Image>();
    im.sprite = Resources.Load<Sprite>("sad");
}
time = 999999f;
}
}

List<float> guarda(List<float> f1)
{
    List<float> f2 = new List<float>();
    for (int i=0; i<f1.Count; i++)
    {
        f2.Add(f1[i]);
    }
    return f2;
}

float DTWDistance(List<float> s, List<float> t)
{
    int n = 1000;
    int m = 1000;
    float cost;
    float[,] DTW = new float[n,m];

    for (int i = 1; i < n; i++)
    {
        DTW[i, 0] = 999999999999f;
    }
    for (int i = 1; i < m; i++)
    {
        DTW[0, i] = 999999999999f;
    }
    DTW[0, 0] = 0;

    for (int i=1; i < n; i++)
    {
        for (int j=1;j < m; j++)
        {
            cost = Math.Abs(s[i]-t[j]);
            //Debug.Log("cost " + cost);
            DTW[i,j]=cost+ Math.Min(DTW[i-1,j], Math.Min(DTW[i,j-1], DTW[i-1,j-1]));
            //Debug.Log("dtw"+DTW[5, 6]);
        }
    }
    return DTW[n-1, m-1];
}

float compara (List<float> s, float a)

```

```

{
    if (a == (float) 1)
    {
        freq=Load("1.txt");
    }
    if (a == (float) 2)
    {
        freq=Load("2.txt");
    }
    if (a == (float) 3)
    {
        freq=Load("3.txt");
    }
    if (a == (float) 4)
    {
        freq=Load("4.txt");
    }
    if (a == (float) 5)
    {
        freq=Load("5.txt");
    }
    return DTWDistance(s, freq);
}

private float[] distancies = new float[5];
private float min = 9999999999999999f;
private int x;

public int Tria(List<float> s)
{
    for (int i = 0; i < 5; i++)
    {
        distancies[i] = compara(s, i + 1);

    }
    for (int h = 0; h < 5; h++)
    {
        if (distancies[h] < min)
        {
            min = distancies[h];
            x = h + 1;
        }
    }
    return x;
}

public List<float> Load(string fileName)

{
    string text = System.IO.File.ReadAllText(fileName);
    string[] split = text.Split('\n');
    //foreach (string k in split);
    List<float> split1 = new List<float>();
    foreach (string k in split)
    {
        float LFOA = 0; /// default value
        if (float.TryParse(k, out LFOA))
        {
            split1.Add(LFOA);
        }
    }
}

```

```

        /// everything is ok
    }
    else
    {
        /// something wrong. vvvvLFO.LFOa has incorrect float value
    }
}
return split1;
}

}

```

## Crear\_txt

```

using UnityEngine;
using System;
using System.IO;

public class crear_txt : MonoBehaviour
{
    public FFTWindow fftwindow;
    public AudioSource b;
    public float[] samples = new float[8192];
    public float time = 0.0f;
    private int para = 0;

    // Use this for initialization
    void Start()
    {
        b = GameObject.Find("Main Camera").GetComponent<

```

```
}
```

## Resultats

```
using UnityEngine;
using System;

public class resultats : MonoBehaviour {
    private TextMesh d,e,f;
    private int x, y, z;
    public GameObject q;
    float amplada = (float)(Screen.width);
    float alcada = (float)(Screen.height);

    // Use this for initialization
    void Start () {
        GameObject.Find("Torna").transform.position = new Vector3(amplada * 0.77f,
    alcada * 0.15f, 0);
        GameObject.Find("Canvas").GetComponent<RectTransform>().sizeDelta = new
    Vector2(amplada, alcada);
        GameObject.Find("Torna").GetComponent<RectTransform>().sizeDelta = new
    Vector2(amplada * 0.1f, alcada * 0.22f);

        d = GameObject.Find("Valor_puntuacio").GetComponent<TextMesh>();
        d.text = Convert.ToString(joc.encertades*5-joc.fallades);
        e = GameObject.Find("Valor_fallades").GetComponent<TextMesh>();
        e.text = Convert.ToString(joc.fallades);
        f = GameObject.Find("Valor_encertades").GetComponent<TextMesh>();
        f.text = Convert.ToString(joc.encertades);
    }

    // Update is called once per frame
    void Update () {

    }

    public void torna()
    {
        Application.LoadLevel("pantalla_principal");
    }
}
```