ANEXO IV, TÍTULO: SCRIPTS DESARROLLADOS PARA LAS INTERACCIONES

gestionaTexto.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class gestionaTexto : MonoBehaviour
{
    public List <string> textoEscribir;
    public TextMeshPro objTexto3D;
    public TextMeshPro objIndicar;
    public float tEntreLetras = 0.8f;
    private bool bIsWriting = false;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (objIsTouched())
        {
            if (Input.touchCount == 1)
            {
                if (Input.GetTouch(0).phase == TouchPhase.Began)
                {
                    introduceTexto();
                }
            }
        }
    }

    private bool objIsTouched()
    {
        foreach (Touch t in Input.touches)
        {
            Ray m_ray = Camera.main.ScreenPointToRay(t.position);
            RaycastHit m_hit;
            if (Physics.Raycast(m_ray, out m_hit, 1000))
            {
                if (m_hit.collider.gameObject == gameObject)
                {
                    return true;
                }
            }
        }
        return false;
    }

    private void introduceTexto()
    {
        if (!bIsWriting )
        {
            objIndicar.gameObject.SetActive(false);
            StartCoroutine(escribeTexto());
        }
    }
```

manipular.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class manipular : MonoBehaviour
{
    public bool puedeRotar = false;
    public bool puedeDesplazarse = false;
    public bool puedeEscalarse = false;
    public float rotSpeed = 0.4f;
    private float initialDistance;
    private Vector3 initialScale;
    private Vector3 finger0Pos;


    void Update()
    {
        if (objIsTouched())
        {
            if (Input.touchCount == 1)
            {
                if (Input.GetTouch(0).phase == TouchPhase.Moved)
                {
                    if (puedeRotar)
                    {
                        this.transform.Rotate(Input.GetTouch(0).deltaPosition.y * rotSpeed,
-Input.GetTouch(0).deltaPosition.x * rotSpeed, 0, Space.World);
                    }
                    if (puedeDesplazarse)
                    {
                        //Vector3 iniPosition = this.transform.position;
                        this.transform.position = finger0Pos;
                        //this.transform.Translate(Input.GetTouch(0).deltaPosition.x, 0,
Input.GetTouch(0).deltaPosition.y, Space.Self);
                    }
                }
            }

            if (Input.touchCount == 2 && puedeEscalarse)
            {
                var touchZero = Input.GetTouch(0);
                var touchOne = Input.GetTouch(1);

                if (touchZero.phase == TouchPhase.Ended || touchZero.phase ==
TouchPhase.Canceled
                   || touchOne.phase == TouchPhase.Ended || touchOne.phase ==
TouchPhase.Canceled)
                {
                    return;
                }

                if (touchZero.phase == TouchPhase.Began || touchOne.phase ==
TouchPhase.Began)
                {
                    initialDistance = Vector2.Distance(touchZero.position, touchOne.position);
                    initialScale = this.transform.localScale;
                }
                else
                {
                    var currentDistance = Vector2.Distance(touchZero.position, touchOne.position);
                    if (Mathf.Approximately(initialDistance, 0)) return;

                    var factor = currentDistance / initialDistance;
                    this.transform.localScale = initialScale * factor;

                }
            }
        }
    }

    private bool objIsTouched()
    {
        foreach(Touch t in Input .touches)
        {
            Ray m_ray = Camera.main.ScreenPointToRay(t.position);
            RaycastHit m_hit;
            if (Physics.Raycast(m_ray, out m_hit, 1000))
            {
                if (m_hit.collider.gameObject == gameObject )
                {
                    finger0Pos = m_hit.point;
                    return true;
                }

            }
        }
        return false;
    }
}
```

pasaFoto.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class pasaFotos : MonoBehaviour
{
    public List<Transform> fotos;
    public float speed = 1.0f;
    private int iCurrentFoto = 0;
    private int iPrevFoto;
    // Start is called before the first frame update
    void Start()
    {

    }

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

    }

    public void cambiaFoto(bool bIsForward)
    {
        iPrevFoto = iCurrentFoto;
        if (bIsForward )
        {
            if (iCurrentFoto == fotos .Count -1)
            {
                iCurrentFoto = 0;
            }
            else
            {
                iCurrentFoto++;
            }
        }
        else
        {
            if (iCurrentFoto == 0)
            {
                iCurrentFoto = fotos.Count - 1;
            }
            else
            {
                iCurrentFoto--;
            }
        }
        if (fotos .Count > 1)
        {
            //hacer pequeña previa
            //hacer grande siguiente
            StartCoroutine(doCambio());
        }
    }

    IEnumerator doCambio()
    {
        yield return StartCoroutine (escalar(Vector3.zero));
        yield return StartCoroutine(escalar(Vector3.one));
    }
    IEnumerator escalar(Vector3 targetEscala)
    {
        if (targetEscala == Vector3 .zero )
        {
            while (fotos [iPrevFoto ].localScale.x > 0)
            {
                fotos[iPrevFoto].localScale -= Vector3 .one * Time .deltaTime * speed;
                yield return null;
            }
            fotos[iPrevFoto].localScale = Vector3.zero;
        }
        else if (targetEscala == Vector3 .one)
        {
            while (fotos[iCurrentFoto].localScale.x < 1)
            {
                fotos[iCurrentFoto].localScale += Vector3.one * Time.deltaTime * speed;
                yield return null;
            }
            fotos[iCurrentFoto].localScale = Vector3.one;
        }
        yield return null;
    }
}
```

quitarCapa.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class quitarCapas : MonoBehaviour
{
    private MeshRenderer[] objQuitar;
    private int index = 0;
    // Start is called before the first frame update
    void Start()
    {
        objQuitar = this.GetComponentsInChildren<MeshRenderer>();
    }

    // Update is called once per frame
    void Update()
    {
        if (index < objQuitar .Length )
        {
            if (objIsTouched())
            {
                if (Input.touchCount == 1)
                {
                    if (Input.GetTouch(0).phase == TouchPhase.Began)
                    {
                        objQuitar[index].gameObject.SetActive(false);
                        index++;
                    }
                }
            }
        }

    }

    private bool objIsTouched()
    {
        foreach (Touch t in Input.touches)
        {
            Ray m_ray = Camera.main.ScreenPointToRay(t.position);
            RaycastHit m_hit;
            if (Physics.Raycast(m_ray, out m_hit, 1000))
            {
                if (m_hit.collider.gameObject == gameObject)
                {
                    return true;
                }
            }
        }
        return false;
    }
}
```

salirConExit.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class salirConExit : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (Input .GetKey (KeyCode.Escape ))
        {
            Application.Quit();
        }
    }
}
```