

Objectiu del projecte

L'objectiu d'aquesta activitat és crear dues animacions programàtiques utilitzant la llibreria DOTween dins d'un projecte Unity, visibles des d'una mateixa càmera, i controlades mitjançant interfície gràfica.

El projecte inclou:

1. Escenari A
 - Un cub realitza un DOJump cap a una posició elevada.
2. Escenari B
 - Dos cubs realitzen una seqüència DOTween composta per *mínim 5 accions* concatenades (*Append*) o sincronitzades (*Join*).
3. UI (User Interface)
 - Un botó Play per iniciar totes les animacions.
 - Un botó Restart per reiniciar-les a l'estat inicial.
4. Build jugable inclosa al repositori.

ESCENARI A – DOJump

L'escenari A conté:

- Un pla a l'esquerra.
- Un cub anomenat CubeA.
- Un punt objectiu elevat (TargetPointA).

Aquest escenari mostra una animació simple utilitzant **DOJump**, que crea un salt amb trajectòria i altura definides.

Codi utilitzat (Escenari A):

```
using UnityEngine;
using DG.Tweening;

Script de Unity (1 referencia de recurso) | 1 referencia
public class ScenarioAController : MonoBehaviour
{
    public Transform cube;
    public Transform targetPoint;

    public float jumpPower = 2f;
    public int numJumps = 1;
    public float duration = 1.5f;

    Vector3 startPosition;
    Tween currentTween;

    Mensaje de Unity | 0 referencias
    void Awake()
    {
        startPosition = cube.position;
    }

    1 referencia
    public void Play()
    {
        if (currentTween != null) currentTween.Kill();

        currentTween = cube
            .DOJump(targetPoint.position, jumpPower, numJumps, duration)
            .SetEase(Ease.OutQuad);
    }

    1 referencia
    public void Restart()
    {
        if (currentTween != null) currentTween.Kill();
        cube.position = startPosition;
    }

    1 referencia
    public void Pause()
    {
        if (currentTween != null) currentTween.Pause();
    }
}
```

ESCENARI B – Seqüència amb Append i Join

L'escenari B conté:

- Un pla a la dreta.
- Dos cubs (CubeB1 i CubeB2).
- Una seqüència complexa amb **mínim cinc accions**:

Accions incloses:

1. Moviment cap endavant
2. Rotació
3. Salt d'un segon cub
4. Moviment addicional
5. Escalat amb loops
6. Tornada a posició inicial

Codi utilitzat (Escenari B)

```
using UnityEngine;
using DG.Tweening;

// Script de Unity (1 referencia de recurso) | 1 referencia
public class ScenarioBController : MonoBehaviour
{
    public Transform cube1;
    public Transform cube2;

    Sequence sequence;
    Vector3 start1, start2;

    // Mensaje de Unity | 0 referencias
    void Awake()
    {
        start1 = cube1.position;
        start2 = cube2.position;
    }

    1 referencia
    public void Play()
    {
        if (sequence != null) sequence.Kill();

        sequence = DOTween.Sequence();

        // 1 - Mover cube1 hacia delante
        sequence.Append(
            cube1.DOMove(cube1.position + Vector3.forward * 1.2f, 0.6f)
        );

        // 2 - Rotación de cube1
        sequence.Append(
            cube1.DORotate(new Vector3(0, 180, 0), 0.5f, RotateMode.LocalAxisAdd)
        );

        // 3 - Salto simultáneo de cube2
        sequence.Join(
            cube2.DOJump(cube2.position + Vector3.up * 1.5f, 1f, 1, 0.8f)
        );

        // 4 - Mover cube2 a la derecha
        sequence.Append(
            cube2.DOMove(cube2.position + Vector3.right * 1f, 0.6f)
        );
    }
}
```

```

// 5 - Escalado simultáneo de cube1
sequence.Join(
    cube1.DOScale(cube1.localScale * 1.3f, 0.4f).SetLoops(2, LoopType.Yoyo)
);

// 6 - Vuelta al inicio de ambos
sequence.AppendInterval(0.2f);
sequence.Append(cube1.DOMove(start1, 0.6f));
sequence.Join(cube2.DOMove(start2, 0.6f));

sequence.OnComplete(() => Debug.Log("Scenario B Complete"));
}

```

1 referencia

```

public void Restart()
{
    if (sequence != null) sequence.Kill();

    cube1.position = start1;
    cube2.position = start2;

    cube1.localScale = Vector3.one;
    cube2.localScale = Vector3.one;

    cube1.rotation = Quaternion.identity;
    cube2.rotation = Quaternion.identity;
}

```

```

public void Pause()
{
    if (sequence != null) sequence.Pause();
}

```

CONTROL GENERAL I BOTONS

Tots dos escenaris són controlats per:

- Botó Play → inicia simultàniament les dues animacions.
- Botó Restart → retorna totes les peces a la posició inicial.

Codi del controlador general

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

Script de Unity (1 referencia de recurso) | 1 referencia
public class DotweenActivityController : MonoBehaviour
{
    public ScenarioAController scenarioA;
    public ScenarioBController scenarioB;

    Mensaje de Unity | 0 referencias
    void Start()
    {
        PauseAll();
    }

    1 referencia
    public void PlayAll()
    {
        scenarioA.Play();
        scenarioB.Play();
    }

    1 referencia
    public void RestartAll()
    {
        scenarioA.Restart();
        scenarioB.Restart();
    }

    1 referencia
    public void PauseAll()
    {
        scenarioA.Pause();
        scenarioB.Pause();
    }
}
```

Codi del controlador de botons

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

El entorno de ejecución Unity crea instancias de este tipo. (Alt+3)
Script de Unity (1 referencia de recurso) | 0 referencias
public class UIController : MonoBehaviour
{
    public Button playButton;
    public Button restartButton;
    public DotweenActivityController controller;

    Mensaje de Unity | 0 referencias
    void Start()
    {
        playButton.onClick.AddListener(() => controller.PlayAll());
        restartButton.onClick.AddListener(() => controller.RestartAll());
    }
}
```