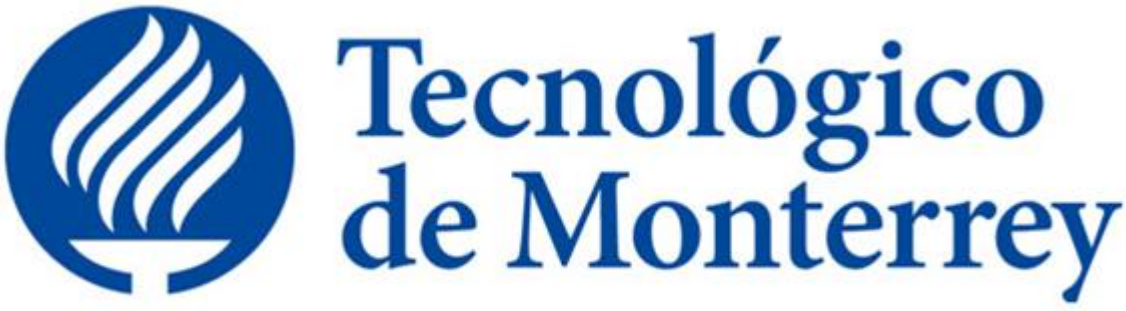


Laboratorio 3

Reporte

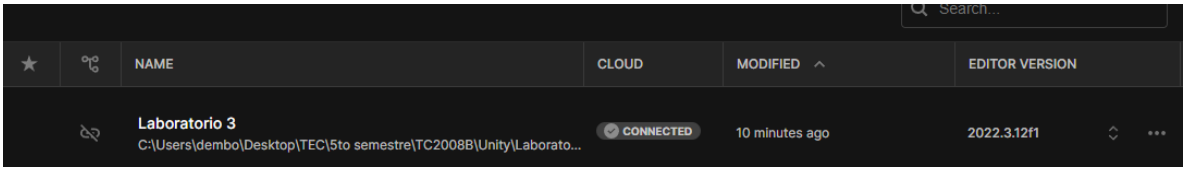
Arturo Díaz López A01709522

TC2008B.301

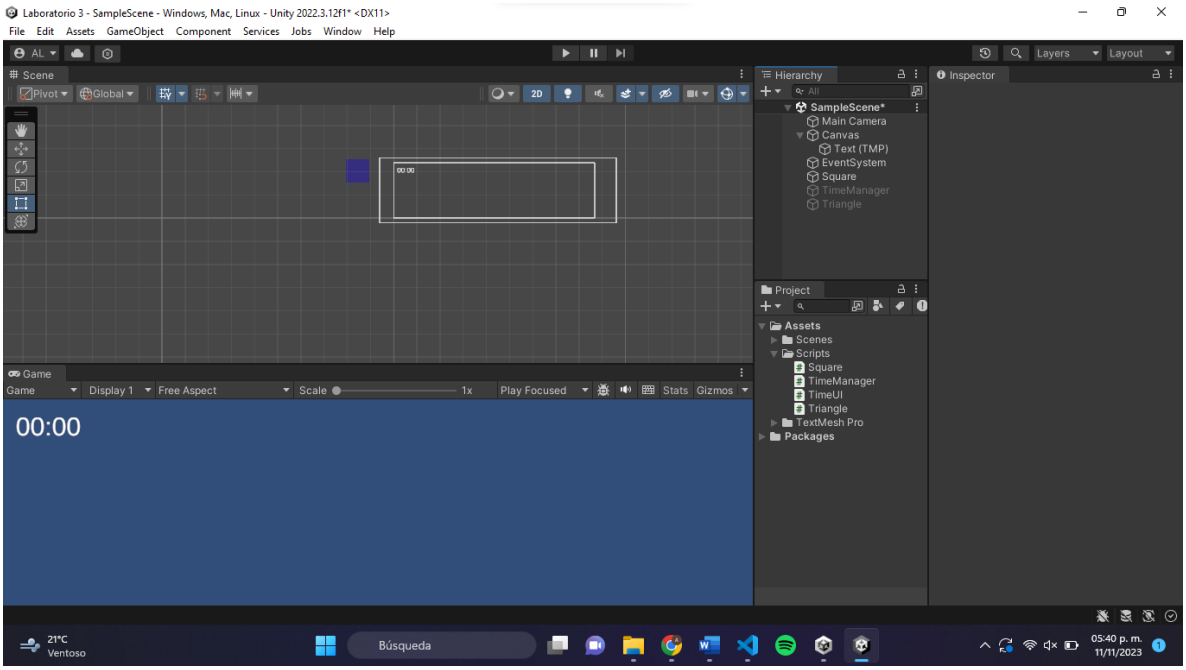
11/11/2023

Video: [Laboratorio 3.mp4]

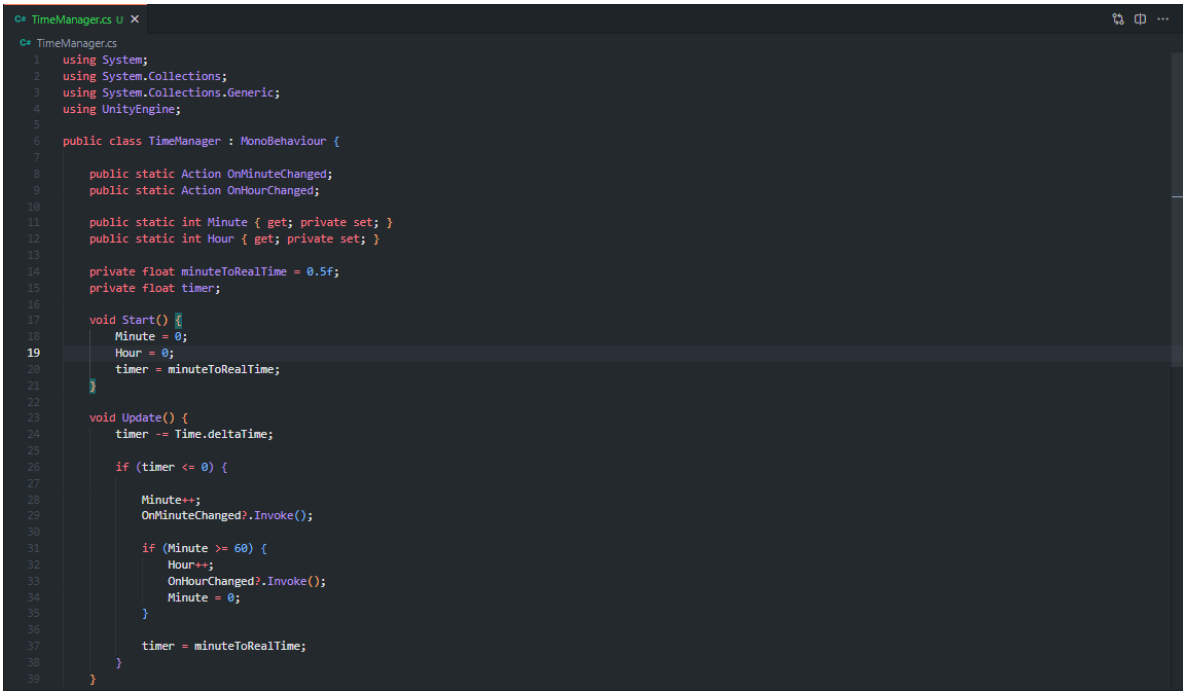
Paso 1



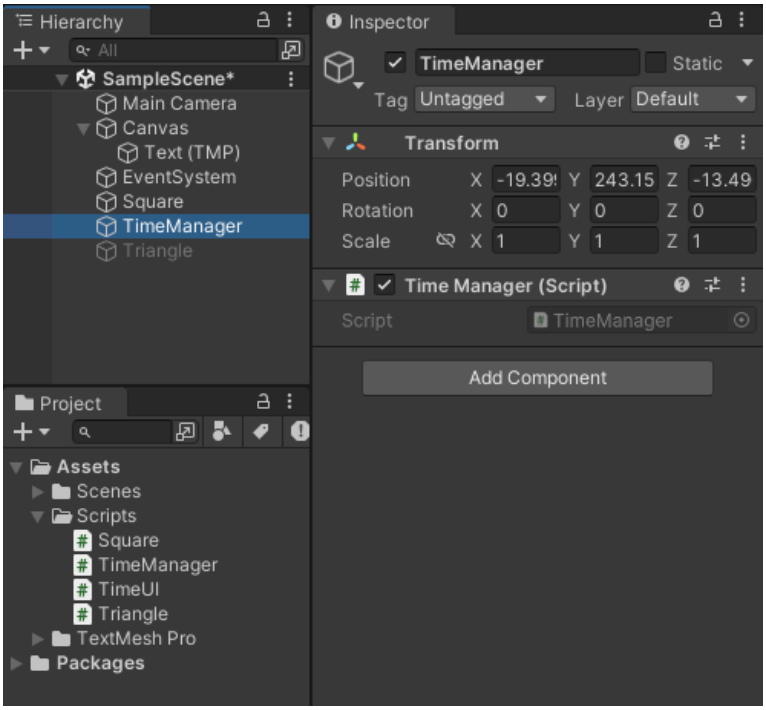
Paso 2



Paso 3


TimeManager.cs U
TimeManager.cs
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class TimeManager : MonoBehaviour {
public static Action OnMinuteChanged;
public static Action OnHourChanged;
public static int Minute { get; private set; }
public static int Hour { get; private set; }
private float minuteToRealTime = 0.5f;
private float timer;
void Start() {
Minute = 0;
Hour = 0;
timer = minuteToRealTime;
}
void Update() {
timer -= Time.deltaTime;
if (timer <= 0) {
Minute++;
OnMinuteChanged?.Invoke();
if (Minute >= 60) {
Hour++;
OnHourChanged?.Invoke();
Minute = 0;
}
timer = minuteToRealTime;
}
}

Paso 4


Hierarchy
All
SampleScene*
Main Camera
Canvas
Text (TMP)
EventSystem
Square
TimeManager
Triangle
Inspector
TimeManager
Static
Tag Untagged
Layer Default
Transform
Position X -19.39 Y 243.15 Z -13.49
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Time Manager (Script)
Script TimeManager
Add Component
Project
Assets
Scenes
Scripts
Square
TimeManager
TimeUI
Triangle
TextMesh Pro
Packages

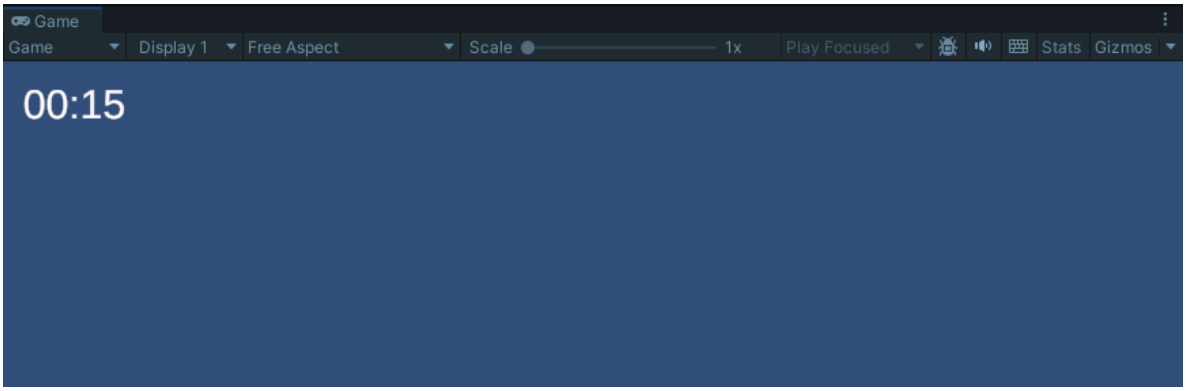
Paso 5

C# TimeUI.cs U

C# TimeUI.cs

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using TMPro;
5
6   public class TimeUI : MonoBehaviour {
7
8       public TextMeshProUGUI timeText;
9
10      void OnEnable() {
11          TimeManager.OnMinuteChanged += UpdateTime;
12          TimeManager.OnHourChanged += UpdateTime;
13      }
14
15      void OnDisable() {
16          TimeManager.OnMinuteChanged -= UpdateTime;
17          TimeManager.OnHourChanged -= UpdateTime;
18      }
19
20      private void UpdateTime() {
21          timeText.text = $"{TimeManager.Hour.ToString("00")}:{TimeManager.Minute:00}";
22      }
23  }
24
```

Paso 6



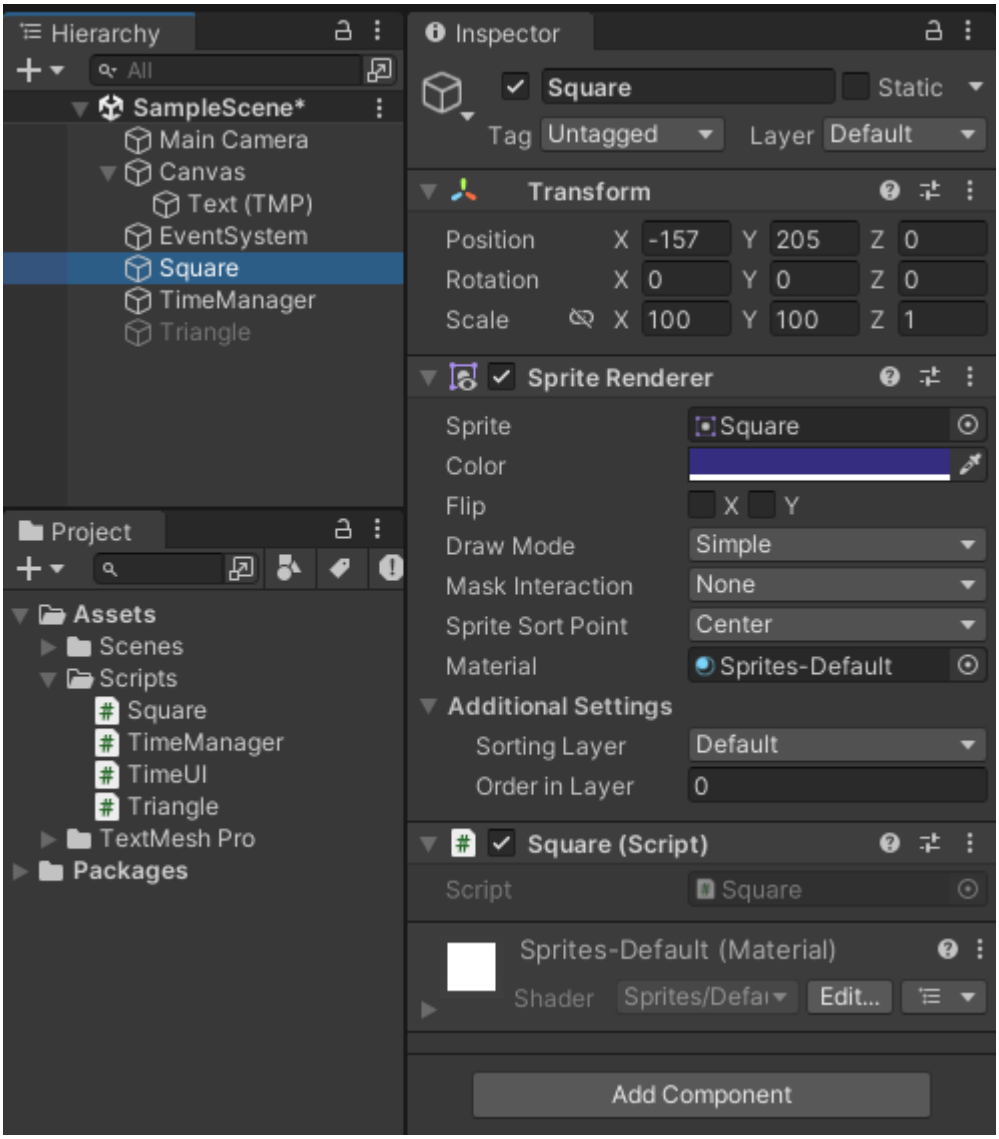
Paso 7

Square.cs U

Square.cs

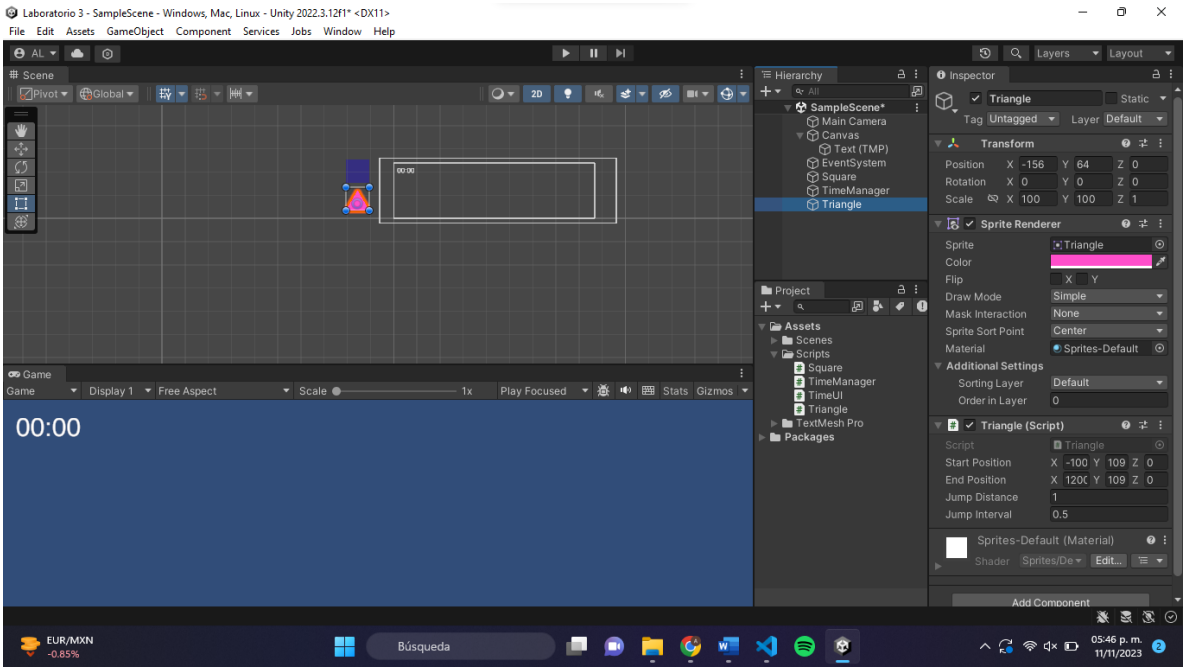
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Square : MonoBehaviour {
    public void OnEnable() {
        TimeManager.OnMinuteChanged += TimeCheck;
    }

    public void OnDisable() {
        TimeManager.OnMinuteChanged -= TimeCheck;
    }

    private void TimeCheck() {
        if(TimeManager.Hour == 00 && TimeManager.Minute == 05)
        {
            StartCoroutine(MoveSquare());
        }
    }

    private IEnumerator MoveSquare() {
        transform.position = new Vector3(-100f,109f,0);
        Vector3 targetPos = new Vector3(1200f,109f,0);

        Vector3 currentPos = transform.position;

        float timeElapsed = 0;
        float timeToMove = 3;

        while(timeElapsed < timeToMove){
            transform.position = Vector3.Lerp(currentPos,targetPos,timeElapsed/timeToMove);
            timeElapsed += Time.deltaTime;
            yield return null;
        }

    }
}

```

Paso 8


Hierarchy
All
SampleScene*
Main Camera
Canvas
Text (TMP)
EventSystem
Square
TimeManager
Triangle
Project
Assets
Scenes
Scripts
Square
TimeManager
TimeUI
Triangle
TextMesh Pro
Packages
Inspector
Square
Static
Tag Untagged
Layer Default
Transform
Position X -157 Y 205 Z 0
Rotation X 0 Y 0 Z 0
Scale X 100 Y 100 Z 1
Sprite Renderer
Sprite Square
Color
Flip X Y
Draw Mode Simple
Mask Interaction None
Sprite Sort Point Center
Material Sprites-Default
Additional Settings
Sorting Layer Default
Order in Layer 0
Square (Script)
Script Square
Sprites-Default (Material)
Shader Sprites/Defa
Edit...
Add Component

Paso 9

Triángulo:

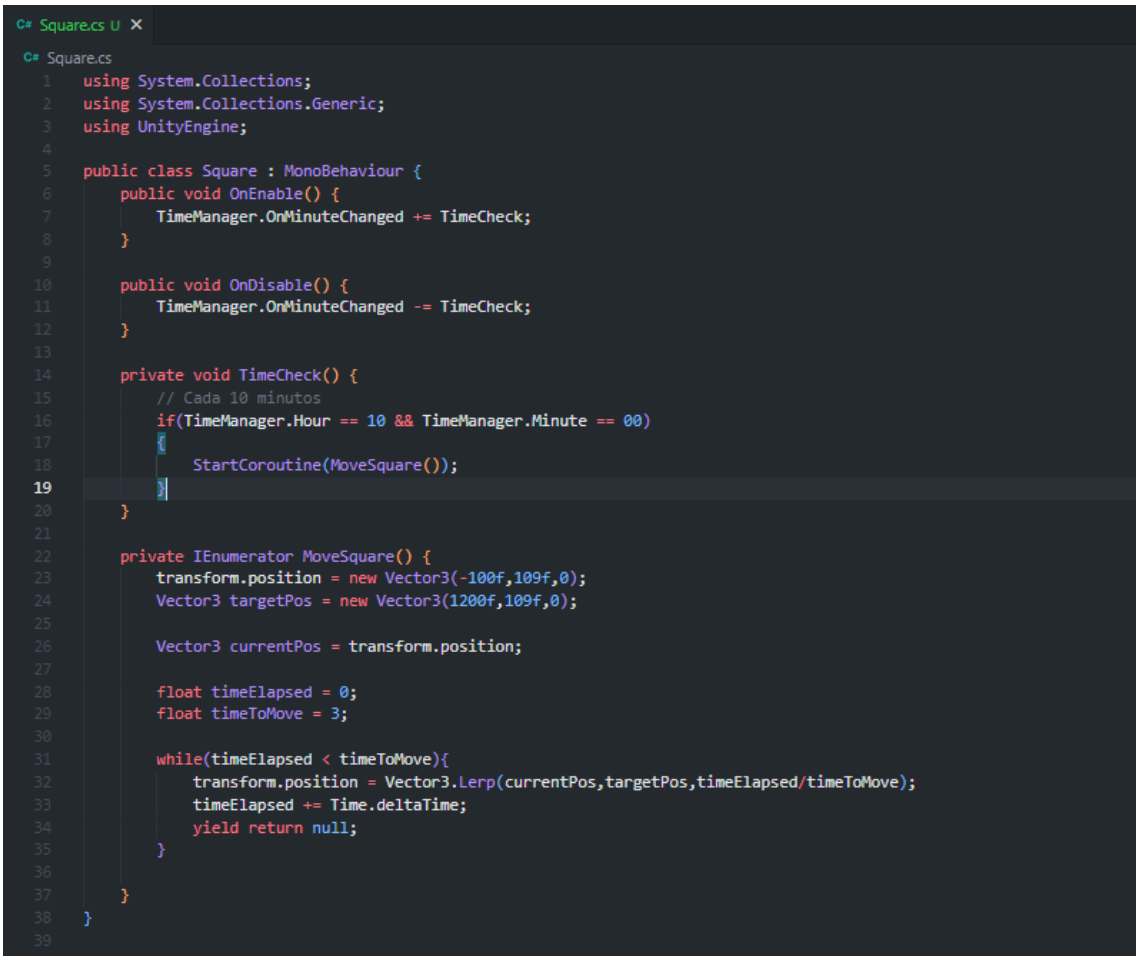


Triangle.cs U

Triangle.cs

```
using System.Collections;
using UnityEngine;

public class Triangle : MonoBehaviour {
    public Vector3 startPosition = new Vector3(-100f, 109f, 0);
    public Vector3 endPosition = new Vector3(1200f, 109f, 0);
    public float jumpDistance = 10000000000000.0f;
    public float jumpInterval = 0.001f;

    private bool shouldMove = false;

    void OnEnable() {
        TimeManager.OnMinuteChanged += TimeCheck;
    }

    void OnDisable() {
        TimeManager.OnMinuteChanged -= TimeCheck;
    }

    void TimeCheck() {
        if (TimeManager.Hour == 0 && TimeManager.Minute == 5) {
            shouldMove = true;
            StartCoroutine(MoveTriangle());
        }
    }

    IEnumerator MoveTriangle() {
        while (shouldMove) {
            float remainingDistance = Vector3.Distance(transform.position, endPosition);

            while (remainingDistance > jumpDistance) {
                transform.position = Vector3.MoveTowards(transform.position, endPosition, jumpDistance*160);
                yield return new WaitForSeconds(jumpInterval);
                remainingDistance = Vector3.Distance(transform.position, endPosition);
            }

            transform.position = endPosition;
            shouldMove = false;

            yield return null;
        }
    }
}
```

Cuadrado se ejecute cada 10 minutos



```
Square.cs U
Square.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Square : MonoBehaviour {
    public void OnEnable() {
        TimeManager.OnMinuteChanged += TimeCheck;
    }

    public void OnDisable() {
        TimeManager.OnMinuteChanged -= TimeCheck;
    }

    private void TimeCheck() {
        // Cada 10 minutos
        if(TimeManager.Hour == 10 && TimeManager.Minute == 00)
        {
            StartCoroutine(MoveSquare());
        }
    }

    private IEnumerator MoveSquare() {
        transform.position = new Vector3(-100f,109f,0);
        Vector3 targetPos = new Vector3(1200f,109f,0);

        Vector3 currentPos = transform.position;

        float timeElapsed = 0;
        float timeToMove = 3;

        while(timeElapsed < timeToMove){
            transform.position = Vector3.Lerp(currentPos,targetPos,timeElapsed/timeToMove);
            timeElapsed += Time.deltaTime;
            yield return null;
        }

    }
}
```

Reflexión final:

Creo que el manejo de tiempo es un concepto de muchísima importancia en el desarrollo de videojuegos. En juegos comunes como FIFA el tiempo es esencial para manejar la duración de un partido, o en Warzone para manejar el encogimiento del círculo seguro para los jugadores. La manera en la que puedo aplicar estos conceptos en mi proyecto es la siguiente:

- Countdowns para lanzar obstáculos al jugador.
- Countdowns para aumentar dificultad de enemigos.
- Tiempo restante antes de enfrentar al jefe final.
- Ciclo día y noche.
- Side quests con contadores de tiempo.

En general, se puede usar para una infinidad de cosas y esto le abre las puertas al dev a dejar volar su imaginación.