

Hitman & the Guy in the Van



Anex 3

29 de maig 2022

Etioipia2

Albert Bueno, Angel Garcia, Juli Ragni



- Annex 3: Codi i Disseny
 - Captures de pantalla amb peus de foto, no es requereix un text explicatiu
 - Disseny de la lògica
 - Fragments de codi, especialment aquell codi complex o que denoti ampli domini tècnic
 - Funcions o classes completes, no línies per separades
 - Codi correctament comentat
 - Disseny del Model de Dades
 - En cas de crear una BD, el seu model E-R
 - En cas de fer servir fitxers, l'estructura de fitxer
 - Implementació del Model de Dades amb fragments de Codi
 - Disseny de les interfícies
 - En cas d'haver fet modelat, rigging o desplegat dels models
 - Wireframe de la interfície
 - Si hi ha interfície web, fragments del css/js. Si hi ha canvas, components del canvas



1 Codi

1.1 Character Manager

```
Script de Unity (7 referencias de recurso) | 3 referencias
public class CharacterManager : NetworkBehaviour
{
    public GameObject currentCharacter;
    public List<GameObject> characters;
    public List<Transform> heads;

    public GameObject cam;
    [HideInInspector] public CinemachineVirtualCamera virtualCamera;

    PCPlayerControllerWithSM playerController;
    Skins current_skin;

    public float transformationTime;

    bool transformed;

    public UnityEvent onStartGuardTransformation;
    public UnityEvent onEndGuardTransformation;

    4 referencias
    public override void OnStartLocalPlayer()
    {
        GameManager.Instance.characterManager = this;

        cam.SetActive(true);
        GetComponent<PCPlayerControllerWithSM>().enabled = true;
        GetComponent<PCPlayerControllerWithSM>().SetCamera(cam);
        GetComponent<AudioListener>().enabled = true;
        GetComponent<FocusCameras>().enabled = true;
        GetComponent<FocusCameras>().ManualStart();

        if (MissionManager.instance != null)
            MissionManager.instance.StartPC(transform);
    }

    Mensaje de Unity | 0 referencias
    private void Awake()
    {
        playerController = GetComponent<PCPlayerControllerWithSM>();
    }
}
```

Classe i atributs





```
1 referencia
public void ChangeCharacter(Skins skin)
{
    currentCharacter.GetComponent<Animator>().Play("Transformarse");

    StartCoroutine(TransformAnimation(skin));
}

1 referencia
IEnumerator TransformAnimation(Skins skin)
{
    StopCoroutine("TransformationDuration");
    playerController.GetComponent<PCPlayerControllerWithSM>().ChangeState(Estado.Interacting, false);
    yield return new WaitForSeconds(4f);
    Transformarse(skin);
}
```

Inici Event Transformació

```
2 referencias
void Transformarse(Skins skin)
{
    playerController.GetComponent<PCPlayerControllerWithSM>().ChangeState(Estado.Standing, false);

    transformed = true;

    if (current_skin != skin)
    {
        print("Transforming to " + skin);

        currentCharacter.SetActive(false);

        GameObject newCharacter;

        switch (skin)
        {
            case Skins.Player:
                newCharacter = characters[1];
                transformed = false;
                virtualCamera.Follow = heads[1];
                break;
            case Skins.Guard:
                newCharacter = characters[2];
                virtualCamera.Follow = heads[2];
                break;
            case Skins.Woman:
                newCharacter = characters[3];
                virtualCamera.Follow = heads[3];
                break;
            case Skins.Man:
                newCharacter = characters[4];
                virtualCamera.Follow = heads[4];
                break;
            case Skins.President:
                newCharacter = characters[5];
                virtualCamera.Follow = heads[5];
                break;
            default:
                newCharacter = characters[0];
                virtualCamera.Follow = heads[0];
                break;
        }
    }
}
```



```
newCharacter.transform.position = currentCharacter.transform.position;
newCharacter.transform.rotation = currentCharacter.transform.rotation;

currentCharacter = newCharacter;

current_skin = skin;

currentCharacter.SetActive(true);

playerController.animator = currentCharacter.GetComponent<Animator>();

this.GetComponent<NetworkAnimator>().animator = currentCharacter.GetComponent<Animator>();
}

if (transformed)
{
    GameManager.Instance.CoolDownTransformation(transformationTime);
    GetComponent<PCPlayerControllerWithSM>().ChangeState(Estado.Standing, true);
    StartCoroutine("TransformationDuration");
}
StartCoroutine("TransformAlert");
}
```

1.2 Estats de NPC'S

```
Script de Unity | 10 referencias
public abstract class TaskMirror : NetworkBehaviour
{
    public string Tname;

    11 referencias
    public abstract void init();

    9 referencias
    public abstract void doTask();

    10 referencias
    public abstract void exit();
}
```

Classe Base



```
Script de Unity (9 referencias de recurso) | 8 referencias
public class TalkMirror : TaskMirror
{
    public delegate void turnToSpeak();
    public turnToSpeak tts;
    public Transform location;
    public Transform talkTo = null;
    List<AudioClip> audios;
    NpcMirror npc;
    public TalkMirror speakTo = null;
    bool doing , otherReady;
    public bool first;
    public int talkPosition = 1;

    [ServerCallback]
    Mensaje de Unity | 0 referencias
    private void Awake()
    {
        npc = this.GetComponent<NpcMirror>();
    }
}
```

Exemple de Task: talk



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

Script de Unity (5 referencias de recurso) | 1 referencia
public class GuardMirror : TaskMirror
{
    NpcMirror npc;
    [SerializeField]
    Transform location;
    public Transform player;
    [SerializeField]
    float rotationY;
    [SerializeField]
    LayerMask enemyLayer;
    bool looking;

    [ServerCallback]
    4 referencias
    public override void init()
    {
        looking = false;
        npc = this.GetComponent<NpcMirror>();
        npc.changeDestinatin(location.position);
    }

    [ServerCallback]
    2 referencias
    public override void doTask()...

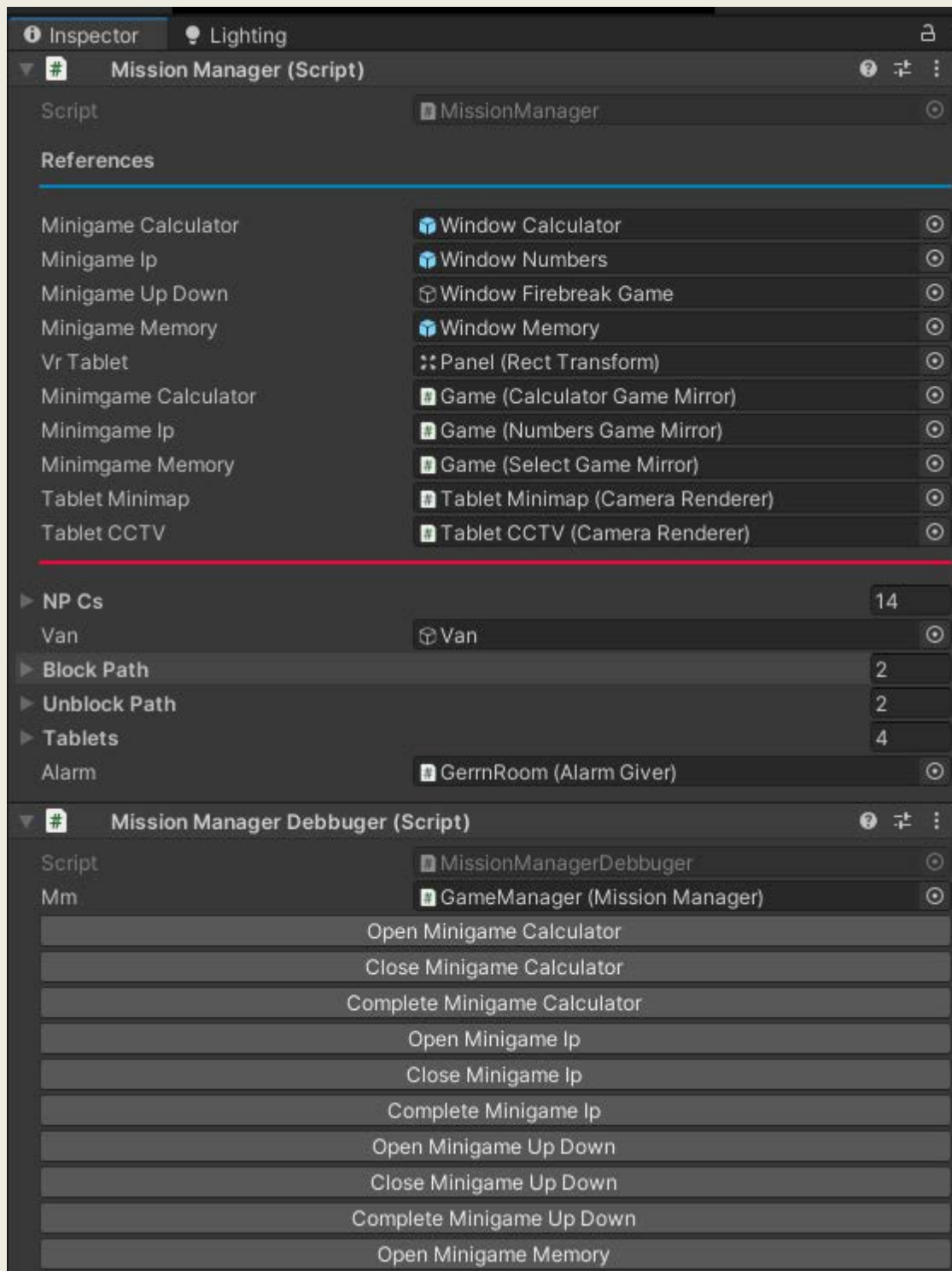
    [ServerCallback]
    3 referencias
    public override void exit()...

    1 referencia
    IEnumerator lookForPlayer()...

    1 referencia
    IEnumerator talk()
    {
        yield return new WaitForSeconds(Random.Range(3,11));
        npc.playAudio(npc.audios.GetAudioClip("Random", Random.Range(0, npc.audios.GetAudioClipCount("Random"))));
    }
}
```



1.3 Mission Manager





1.4 Mirror

```
[Command(requiresAuthority = false)]
```

4 referencias | Añadido by angelgamo162@gmail.com on sábado, 28 de mayo de 2022

```
void Move(int move)
{
    MoveVisual(move);
    if (isClient)
        return;

    index += move;
    if (index < 0)
        index += parent.childCount;
}
```

```
[ClientRpc]
```

2 referencias | Añadido by angelgamo162@gmail.com on sábado, 28 de mayo de 2022

```
void MoveVisual(int move)
{
    for (int i = 0; i < 4; i++)
    {
        int a = (index + i) % parent.childCount;
        numbers[a].color = Color.white;
    }
    index += move;
    if (index < 0)
        index += parent.childCount;
    for (int i = 0; i < 4; i++)
    {
        int a = (index + i) % parent.childCount;
        numbers[a].color = Color.red;
    }
}
```



2 Format Dades

```
[Serializable]
0 referencias
public class GameData
{
    public float tiempo;
    public string pcName;
    public string vrName;
    public int vecesPillado;
    public int erroresMini;
    public int asseginatos;
    public int convertido;
    public int vecesAgua;

    0 referencias
    public GameData(float tiempo, string pcName, string vrName, int vecesPillado, int erroresMini, int asseginatos, int convertido, int vecesAgua)
    {
        this.tiempo = tiempo;
        this.pcName = pcName;
        this.vrName = vrName;
        this.vecesPillado = vecesPillado;
        this.erroresMini = erroresMini;
        this.asseginatos = asseginatos;
        this.convertido = convertido;
        this.vecesAgua = vecesAgua;
    }
}
```

Dades a Guardar

```
0 referencias
public static void SaveToJson<T>(T toSave, string filename)
{
    string content = JsonUtility.ToJson(toSave);
    WriteFile(GetPath(filename), content);
}

1 referencia
public static List<T> ReadListFromJson<T>(string filename)
{
    string content = ReadFile(GetPath(filename));

    if (string.IsNullOrEmpty(content) || content == "{}")
    {
        return new List<T>();
    }

    List<T> res = JsonHelper.FromJson<T>(content).ToList();

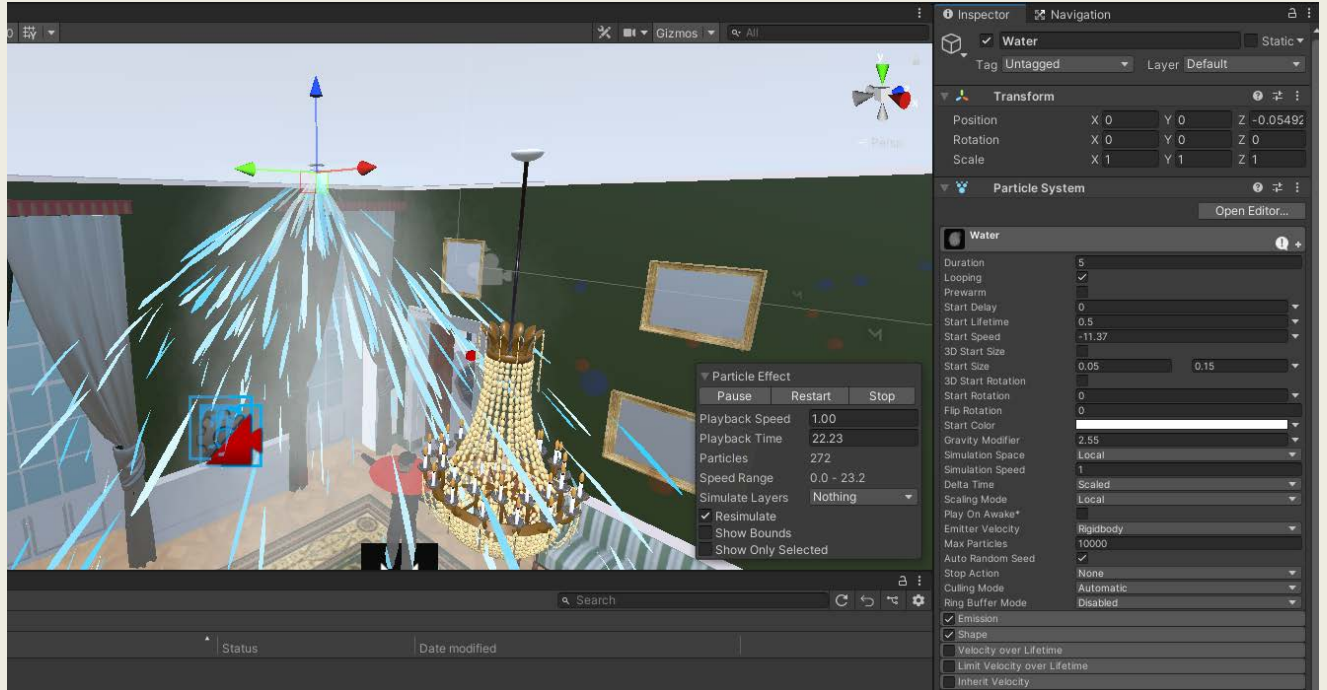
    return res;
}
```

Writer i Reader de JSON



3 Gràfics

3.1 Particle System



Efecte d'aigua a l'alarma anti incendis



3.2 UI

```
Script de Unity (3 referencias de recurso) | 1 referencia
public class TransformCooldown : MonoBehaviour
{
    public Image image;

    1 referencia
    public void Cooldown(float time)
    {
        StopAllCoroutines();
        StartCoroutine(CooldownCorrutine(time));
    }

    1 referencia
    IEnumerator CooldownCorrutine(float time)
    {
        image.fillAmount = 1;
        while(image.fillAmount > 0)
        {
            image.fillAmount -= 1 / time * Time.deltaTime;
            yield return new WaitForEndOfFrame();
        }
    }
}
```

Cooldown de la Transformació