****

**IES AUGUSTO GONZÁLEZ DE LINARES**

**DEPARTAMENTO DE INFORMÁTICA**

**Anexo I**

**Requisitos PMDM**

PROYECTO DICIEMBRE 2023 2º CURSO

GRADO SUPERIOR DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA

2022/2023

Índice

[1. Documentación PMDM 2](#_Toc153042232)

# Documentación PMDM

## **Generación de GDD del videojuego incluido en README del GitLab, donde se explicitarán las características principales del juego.**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamentePartimos del GDD que diseñamos para la asignatura ya que en este reto vamos a continuar con el desarrollo de este, lo primero fue subir nuestro repositorio de GitHub en GitLab, esto se puede hacer si alguien tiene clonado el repositorio en su ordenador y establece múltiples remotos, y actualizar el archivo “*readme.md*”.

Software 1 – SourceTree con múltiples remotos

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Web 1 – Repositorio Remoto GitLab

## **Animaciones 2D o 3D, ya sea de personaje o de algún elemento del escenario.**

El Script CharacterMovement se ocupa de controlar el movimiento del personaje utilizando los Input Axis de Unity.

Captura de pantalla de computadora

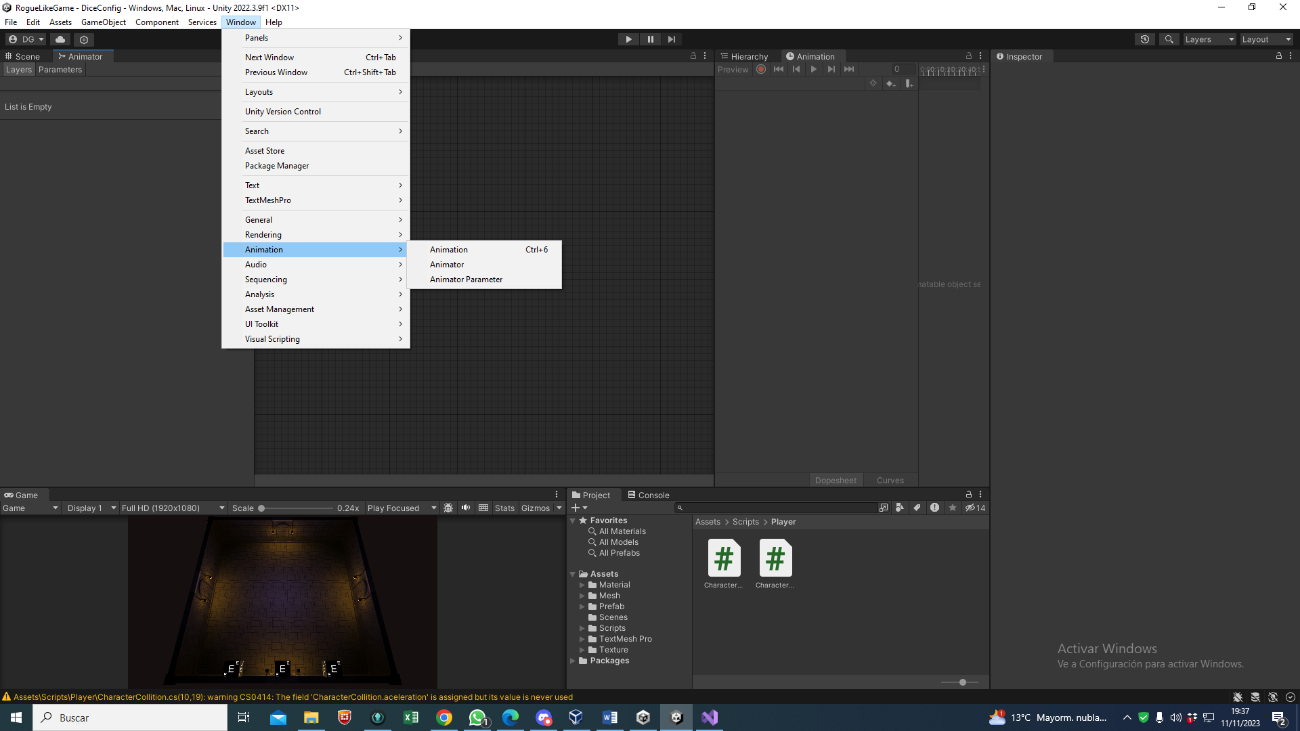
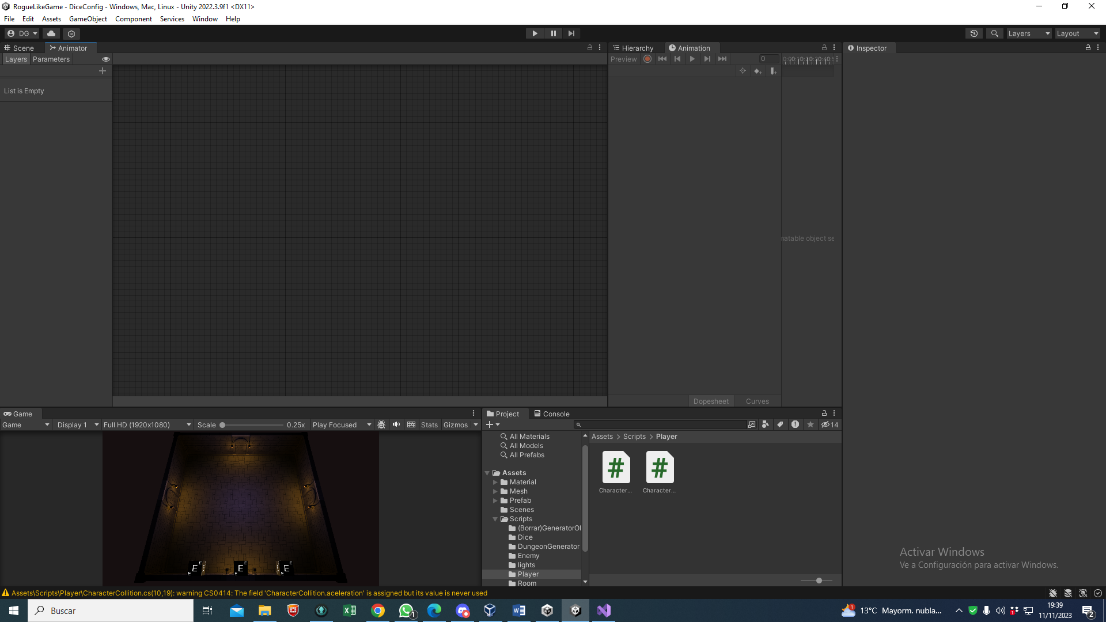
Descripción generada automáticamenteAl mover al personaje sobre el espacio esta ira rotando de forma progresiva hasta que apunte en la dirección en la que se mueve.

Código 1 - Script CharacterMovement

Unity cuenta con su gestor de animaciones el cual permite montar secuencias de animación de Objetos.

En lo que llevamos realizado de proyecto no hemos podido implementar aun las animaciones para el personaje principal, pero llegaremos a ello en el futuro.

Software 2 – Unity Animator View



## **Incluirá los elementos ya vistos en el módulo hasta el momento: lógica de juego, inclusión de asset (sprites o modelos 3D), sonido, efectos…**

Pantalla de computadora

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica

Descripción generada automáticamentePara desarrollar este videojuego empezamos con assets básicos, investigamos los múltiples assets que soporta Unity y buscamos assets de libre uso para hacer una versión básica del juego.

Software 3 – Assets Unity

Software 4 – Escena con Assets Free Use

## **Varias escenas correspondientes con: escena de inicio, varios niveles de juego (al menos uno por integrante del equipo), escena de fin.**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamentePara el desarrollo del juego hemos usado múltiples escenas para programar las diferentes funciones, aunque en el juego final solo hay tres, el menú principal, el selector de datos para jugar, y la mazmorra aleatoria.

Interfaz de usuario gráfica

Descripción generada automáticamente

Software 5 – Explorador de Archivos en Unity

Software 6 – Build Settings del proyecto

## **Pruebas y optimización: uso del profiler de Unity y realización de pruebas de usuario.**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteA lo largo de desarrollo a sido necesario ir optimizando el juego para que pueda ser ejecutado en cualquier ordenador, es comprobar el número de vértices de los assets 3D que nos descargamos, a menor número de vértices mejor rendimiento ya que el ordenador no tiene que procesar tantas coordenadas para mostrar un objeto.

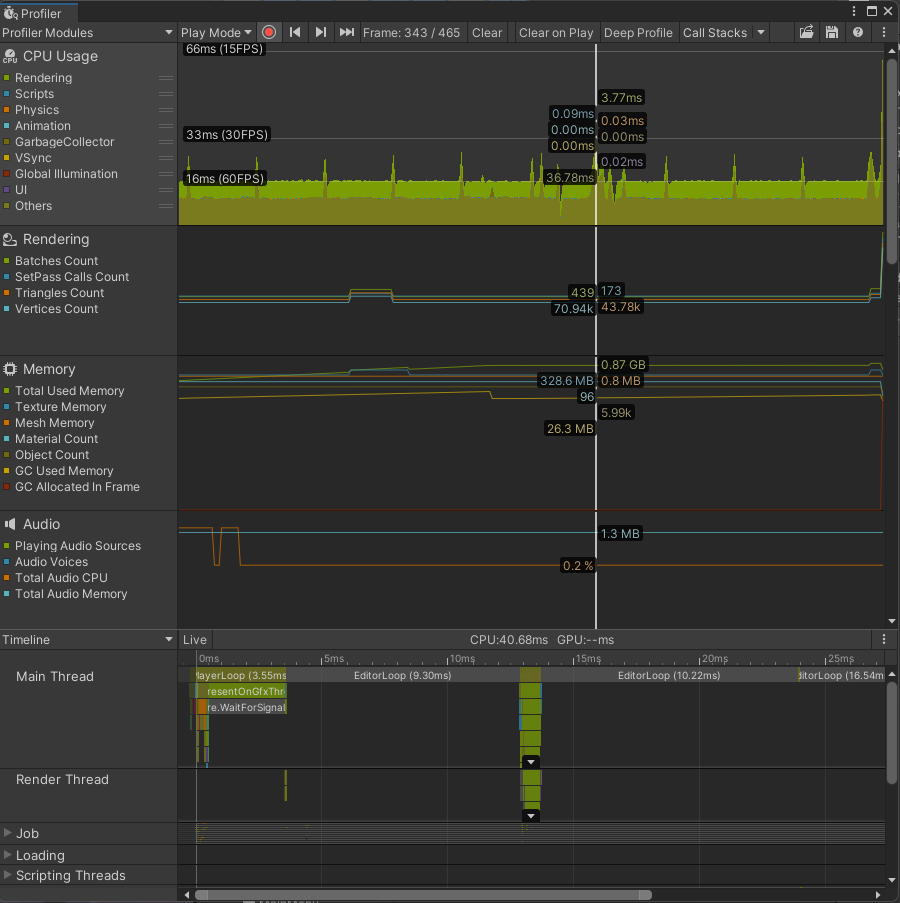
Software 7 – Inspector Unity (Mesh)

Pero muchas veces es necesario observar en que esta empleando el ordenador los recursos para optimizar las partes que se demoren mas de lo necesario, por ejemplo, que un script se quede bloqueado un par de fotogramas hace que el rendimiento empeore, para verlo de forma sencilla hay una paquete hecho por Unity llamado *“Profiler”*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Software 8 – Package Manager(Profiler)



Software 9 – Modulo Profiler

Otra forma de optimización es mirando que codigo esta *“muerto”*, es decir que nunca es ejecutado, eliminar este codigo residual hace que el compilador tarde unos milisegundos menos en compilar, parece que no se obtiene una gran mejoría ya que nuestro juego no tiene muchas líneas de codigo, pero en juegos mucho mas grande esto puede suponer que el juego sea injugable. Para ello unity dispone del paquete llamado *“Code Coverage”*.

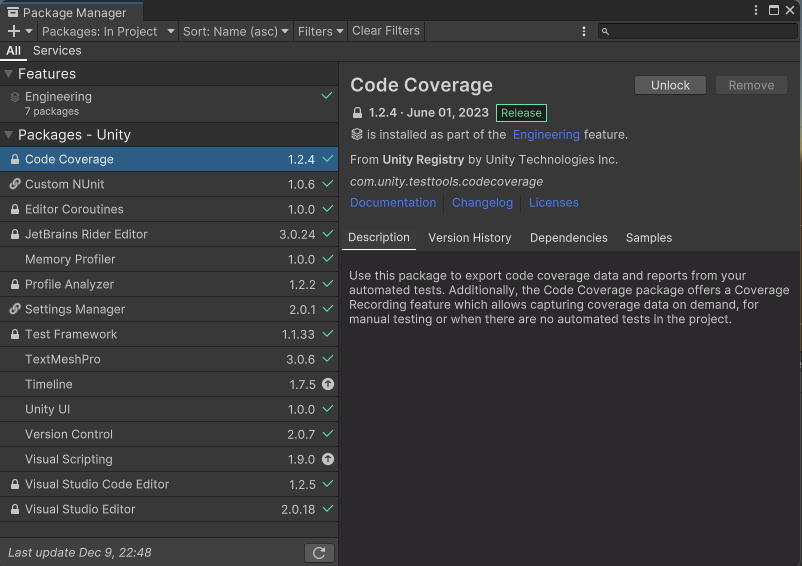


Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza media

Web 2 – Test Code Coverage

Software 10 – Package Manager (Code Coverage)

Además, otra buena forma de probar el juego es realizar pruebas con usuarios que no sean del desarrollo así puedes encontrar errores que los desarrolladores no encuentran por la visión sesgada.

## **Documentación: se generará tanto la documentación técnica como el manual de usuario del juego.**

## **Generación ejecutable: generar ejecutable para Windows (.exe).**

## **Generación APK: el proyecto debe incluir la posibilidad de generar el juego para dispositivos móviles Android, el control deberá ajustarse para utilizar entradas táctiles.**

## **Utilización de Assets: pueden utilizarse asset gratuitas del store de Unity o externas siempre de acuerdo con la licencia de uso que tengan.**

## **Creación de un instalador para el juego**