
Framework de comportamientos de enemigos
para videojuegos 2D
An enemy behaviour framework for 2D
videogames



Trabajo de Fin de Grado
Curso 2024–2025

Autor

Francisco Miguel Galván Muñoz
Cristina Mora Velasco

Director

Guillermo Jimenez Díaz

Colaborador

Guillermo Jimenez Díaz

Grado en **Desarrollo de Videojuegos**

Facultad de Informática

Universidad Complutense de Madrid

Framework de comportamientos de
enemigos para videojuegos 2D
An enemy behaviour framework for 2D
videogames

Trabajo de Fin de Grado en **Desarrollo de Videojuegos**

Autor

Francisco Miguel Galván Muñoz
Cristina Mora Velasco

Director

Guillermo Jimenez Díaz

Colaborador

Guillermo Jimenez Díaz

Convocatoria: *Junio 2025*

Grado en Desarrollo de Videojuegos

Facultad de Informática

Universidad Complutense de Madrid

13 de Junio de 2025

Dedicatoria

*A nuestras gatas por inventar el arte de
convertir el dolor de un arañazo en la alegría
de un beso*

*Y a nuestras familias por el apoyo
incondicional mostrado en la consecución de
este trabajo*

Agradecimientos

A Guillermo por ser el mejor tutor de Trabajo de Fin de Grado posible y por habernos dado la oportunidad de cumplir un sueño al llegar a la consecución de este grado. A todo el resto del profesorado que nos ha enseñado tantas cosas y nos han acompañado estos años. Y sobre todo a nuestras familias que día tras día han estado con nosotros, ayudándonos a crecer, apoyándonos incondicionalmente y dándonos fuerzas para seguir.

Resumen

Framework de comportamientos de enemigos para videojuegos 2D

El arte de hacer un videojuego es la combinación homogénea de varias artes para crear una experiencia interactiva. Entre estas partes encontramos arte, sonido, programación y diseño. A la hora de hacer el arte, el ser humano siempre se ha ayudado de herramientas las cuales ha fabricado él mismo y le ha ayudado a llevar a cabo las tareas de manera más eficaz y precisa. A la hora de hacer un videojuego de cualquier tipo, el equipo detrás de la obra tiene que dejar clara dos ideas al jugador: el objetivo del juego y los obstáculos a los que se va a tener que enfrentar para alcanzarlo. Haciendo especial hincapié en el tipo de juegos analizados en este trabajo, el principal obstáculo serán enemigos, y para que estos supongan un desafío para el jugador deberán contar con una inteligencia artificial acorde al diseño del mismo.

El objetivo de este trabajo es crear una herramienta cuyo propósito sea generar enemigos para videojuegos de plataformas 2D. Para ello se creará un catálogo de componentes para Unity con los que poder crear, sin tener conocimientos previos de programación, inteligencias artificiales de enemigos. Este catálogo de componentes es el resultado de nuestro trabajo de investigación y posterior abstracción de comportamientos de enemigos en juegos de plataformas 2D.

Palabras clave

Inteligencia Artificial, IA, Unity, Enemigo, Maquinas de Estado, Estado, Sensor

Abstract

An enemy behaviour framework for 2D videogames

The art of making a video game is the homogeneous combination of various arts to create an interactive experience. Among these parts we find art, sound, programming and design. When it comes to making art, humans have always used tools which they have made themselves and which have helped them to carry out tasks more efficiently and accurately. When making a video game of any kind, the team behind the work has to make two ideas clear to the player: the objective of the game and the obstacles that the player will have to face to achieve it. With special emphasis on the type of games analyzed in this work, the main obstacle will be enemies, and for them to be a challenge for the player they must have an artificial intelligence according to the game design.

The objective of this work is to create a tool whose purpose is to generate enemies for 2D platform video games. For this purpose, a catalog of components for Unity will be created with which to create, without previous programming knowledge, enemy artificial intelligences. This catalog of components is the result of our research work and subsequent abstraction of enemy behaviors in 2D platform games.

Keywords

Artificial Intelligence, AI, Unity, Enemy, State Machine, State, Sensor

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Plan de trabajo	2
2. Estado de la Cuestión	5
2.1. Análisis herramientas	5
2.2. Técnicas	5
2.2.1. Maquinas de estado finitas	5
2.2.2. Behaviour Bricks	5
2.3. Motores de videojuegos	5
2.3.1. Unity	5
2.3.2. Unreal	5
2.3.3. Godot	5
2.3.4. GameMaker	5
2.3.5. Construct 3	5
2.4. Conclusiones	5
3. Descripción del Trabajo	7
3.1. Contexto	7
3.1.1. Enemigo	7
3.2. Composición	7
3.2.1. Máquina de estados finita	8
3.3. Ejemplos	8
4. Conclusiones y Trabajo Futuro	11
Introduction	13
Conclusions and Future Work	15
Contribuciones Personales	17

A. Título del Apéndice A	19
B. Título del Apéndice B	21

Índice de figuras

1.1. Tabla de Gantt	4
3.1. Enemigo General	8
3.2. Ejemplo de imagen	9

Índice de tablas

3.1. Tabla de ejemplo	9
---------------------------------	---

Introducción

*“Los seres humanos no nacen para siempre el día en que sus
madres los alumbran, sino que la vida los obliga a parirse a
sí mismos una y otra vez”*
— Gabriel García Márquez

1.1. Motivación

A lo largo de los años, los videojuegos han evolucionado considerablemente, llegando a convertirse en una parte muy importante del entretenimiento de hoy en día. Desde las sencillas primeras experiencias de las consolas hasta los complejos mundos actuales, los videojuegos han conquistado audiencias de todas las edades y culturas. A lo largo de esta evolución los enemigos han jugado un papel fundamental, haciendo que los videojuegos presentasen desafíos que atrapan a los jugadores. En el caso de los videojuegos 2D de plataformas, los enemigos son más que una simple oposición del jugador si no que son clave para mostrar la esencia del juego. El Pac-Man no sería lo mismo sin sus fantasmas ni el Super Mario Bros sin sus Goombas.

Diseñar enemigos, especialmente en mundos plataforma en los cuales los enemigos tienen un papel tan importante es siempre una tarea compleja. No solo trata de darles cierta apariencia si no que tienen que tener unos comportamientos y características únicas. Esto implica que la persona encargada de esta tarea tiene que tener ciertos conocimientos en arte, diseño y programación. En la actualidad, este trabajo es realizado por varias personas. Teniendo como mínimo tres puestos: programador, diseñador y artista.

En los últimos años han surgido herramientas que simplifican significativamente el trabajo de los diseñadores, pero muy pocas están centradas específicamente en el diseño de enemigos. Teniendo como resultado que no sea necesario que adquiriera conocimientos sobre programación.

1.2. Objetivos

Este trabajo tiene como objetivo principal el diseño y desarrollo de un herramienta en C sharp para el motor de videojuegos Unity, que simplifique y agilice el

proceso de creación de enemigos en mundos 2D de plataformas. La herramienta contará con un catalogo de comportamientos fácil de manejar para cualquier persona independientemente de sus conocimientos de programación. El catálogo estará compuesto por tres categorías diferentes: acciones, sensores y eventos. Estos se definirán más adelante.

Para realizar el trabajo se realizará un estudio sobre el tema a tratar, herramientas y motores ya existentes. Así como una conclusión sobre el análisis.

1.3. Plan de trabajo

Para llevar a cabo este trabajo, se ha seguido una metodología ágil, Scrum. Esta metodología, permite crear un flujo de trabajo enfocado en la iteración y continua mejora, asegurando un avance en el desarrollo eficiente y posibles adaptaciones frente a problemas detectados durante el proceso. El trabajo se dividirá en cuatro bloques: investigación y planificación, desarrollo de la memoria, desarrollo de la herramienta y pruebas con usuarios. Cada bloque a su vez se dividirá entres subsecciones explicadas a continuación.

- Investigación y planificación:
 - Estudio del problema: En esta primera fase se realizará un estudio del estado del arte, centrado en el papel de los enemigos en los videojuegos, su importancia en la jugabilidad y las diferentes técnicas utilizadas para su diseño y comportamiento.
 - Selección y estudio de herramientas: Esta fase implicará un análisis comparativo de distintas técnicas y motores de videojuegos evaluando sus ventajas y desventajas, así como un estudio de su funcionamiento y arquitecturas.
 - Diseño de la herramienta: En esta etapa, se definirá la arquitectura de la herramienta propuesta, describiendo las técnicas empleadas, esquemas de funcionamiento y organización de elementos principales.
- Desarrollo de la memoria:
 - Redacción inicial: Es esta fase del trabajo se procederá a la redacción inicial de los contenidos cubriendo todos los puntos especificados en el índice.
 - Revisión y corrección: Una vez completada la redacción inicial, se realizarán las correcciones necesarias tras revisar exhaustivamente el documento.
 - Conclusiones y trabajo futuro: Tras finalizar los desarrollos y las pruebas de usuario, se redactarán las conclusiones obtenidas en base a los resultados y se detallaran los posibles pasos a seguir en un futuro.
- Desarrollo de la herramienta:

- Implementación de funcionalidades principales: En esta etapa se implementarán las funcionalidades principales de los movimientos básicos incluyendo la integración con sensores y emisores permitiendo la interacción entre ellos.
 - Implementación ayuda visual: Se desarrollaran ayudas visuales destinadas a servir como referencias para los diseñadores, incluyendo elementos gráficos que faciliten la comprensión de los comportamientos. Además se implementará una herramienta que permita elegir y configurar de una forma más sencilla los componentes.
 - Pruebas y depuración: Se llevará a cabo un proceso iterativo de pruebas que aseguren la funcionalidad de la herramienta, corrigiendo los errores detectados durante su implementación.
- Pruebas con usuarios:
- Primera fase de pruebas: Se harán pruebas con gente que no haya probado la herramienta antes, siguiendo un plan de pruebas especificado en el apartado ".evaluación con usuarios". Las pruebas estarán centradas en: detectar posibles errores en las funcionalidades principales, validar la funcionalidad y evaluar la usabilidad y claridad.
 - Segunda fase de pruebas: Tras implementar mejoras recibidas del feedback de la primera prueba, se llevará a cabo una segunda prueba de verificación.
 - Corrección y resultados: Tras analizar los resultados de cada fase de prueba, se documentarán los errores y dificultades encontradas. Tras esto se procederá a implementar las correcciones necesarias para mejorar los resultados.

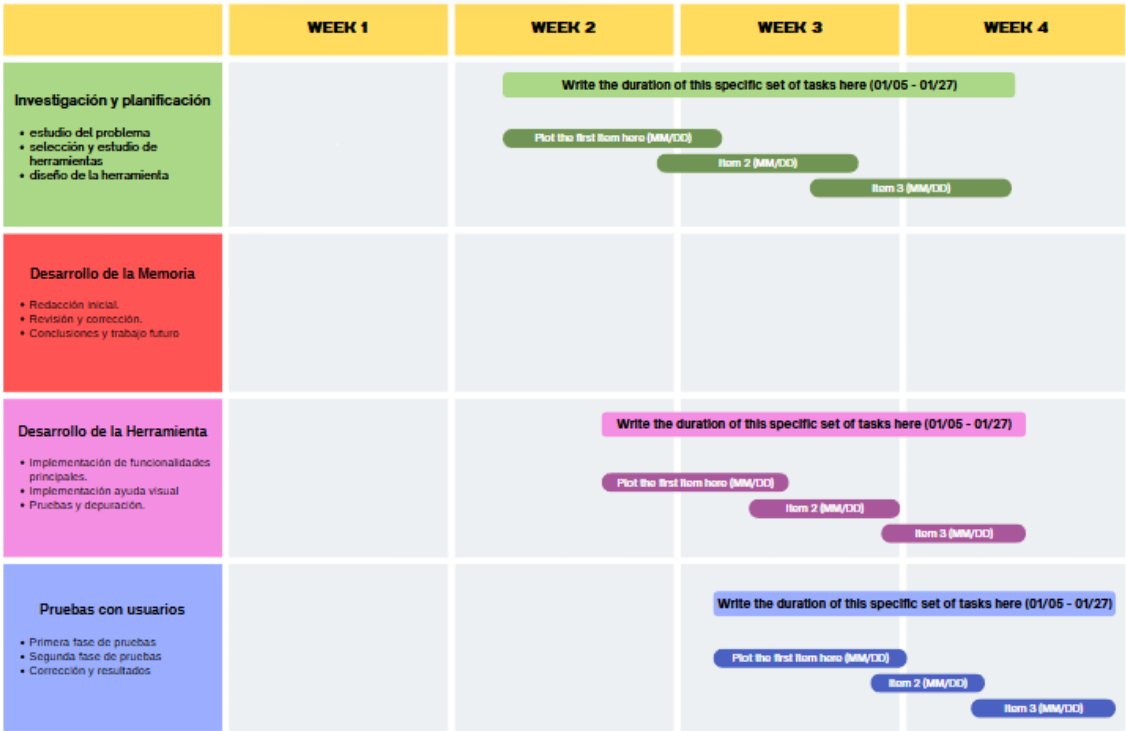


Figura 1.1: Tabla de Gantt

Capítulo 2

Estado de la Cuestión

Este capítulo consta de un análisis de las herramientas sobre generación de enemigos ya existentes y de las técnicas que se usarán. Por último tendrá un análisis de los motores de videojuegos más fáciles de utilizar y unas conclusiones sobre el tema.

2.1. Análisis herramientas

2.2. Tecnicas

2.2.1. Maquinas de estado finitas

2.2.2. Behaviour Bricks

2.3. Motores de videojuegos

2.3.1. Unity

2.3.2. Unreal

2.3.3. Godot

2.3.4. GameMaker

2.3.5. Construct 3

2.4. Conclusiones

En el estado de la cuestión es donde aparecen gran parte de las referencias bibliográficas del trabajo. Una de las formas más cómodas de gestionar la bibliografía en \LaTeX es utilizando **bibtex**. Las entradas bibliográficas deben estar en un fichero con extensión *.bib* (con esta plantilla se proporciona el fichero *biblio.bib*, donde están las

entradas referenciadas más abajo). Cada entrada bibliográfica tiene una clave que permite referenciarla desde cualquier parte del texto con los siguiente comandos:

- Referencia bibliografica con cite: ?
- Referencia bibliográfica con citep: (?)
- Referencia bibliográfica con citet: ?

Es posible citar más de una fuente, como por ejemplo (???)

Después, L^AT_EX se ocupa de rellenar la sección de bibliografía con las entradas **que hayan sido citadas** (es decir, no con todas las entradas que hay en el .bib, sino sólo con aquellas que se hayan citado en alguna parte del texto).

Bibtex es un programa separado de latex, pdf_latex o cualquier otra cosa que se use para compilar los .tex, de manera que para que se rellene correctamente la sección de bibliografía es necesario compilar primero el trabajo (a veces es necesario compilarlo dos veces), compilar después con bibtex, y volver a compilar otra vez el trabajo (de nuevo, puede ser necesario compilarlo dos veces).

Capítulo 3

Descripción del Trabajo

En este capítulo se describe el framework de enemigos creado, mediante el diseño de componentes más sencillos. Primero se describirá el contexto y por tanto la utilidad de la herramienta. Después se explicará que elementos la componen y por último se detallarán algunos ejemplos de uso.

3.1. Contexto

3.1.1. Enemigo

Los enemigos son entidades programadas para oponerse al jugador y crear desafíos dentro del juego. Generalmente, tienen características, comportamientos y habilidades diseñadas para interactuar con las mecánicas del juego y contribuir a la experiencia del jugador.

3.2. Composición

En este trabajo, se ha decidido entender como enemigo a cualquier entidad que pueda repercutir de forma negativa en el jugador. Esto significa que no se limita el concepto de enemigo a figuras típicas, como monstruos o soldados hostiles, sino que se amplía su definición a toda entidad que suponga un riesgo, dificultad o amenaza para el progreso o el bienestar del jugador dentro del juego, como pueden ser pinchos, lava o gotas de ácido. Además separamos cada enemigo por comportamientos diferentes, implicando que elementos que clásicamente aparecen en conjunto, como la tubería y la gota de ácido o la bala y el pistolero, serán considerados como dos enemigos distintos.

3.2.1. Máquina de estados finita

La Máquina de Estados Finita (FSM, Finite State Machine) es el núcleo de la lógica que define el comportamiento de los enemigos en nuestro diseño de videojuegos. Cada enemigo tiene su propia FSM, configurada específicamente para representar sus patrones de acciones, reacciones y relaciones en el juego. La FSM organiza el comportamiento de los enemigos mediante estados y transiciones: Los estados agrupan las acciones que el enemigo puede realizar en un momento dado. Las transiciones permiten cambiar de un estado a otro y son activadas por sensores. Estos conceptos (estados, sensores y eventos) se desarrollan con mayor detalle en los apartados siguientes.

El objeto enemigo se define mediante una máquina de estados finita y un script que contiene información relevante. A continuación, se describen ambos conceptos y sus propiedades.

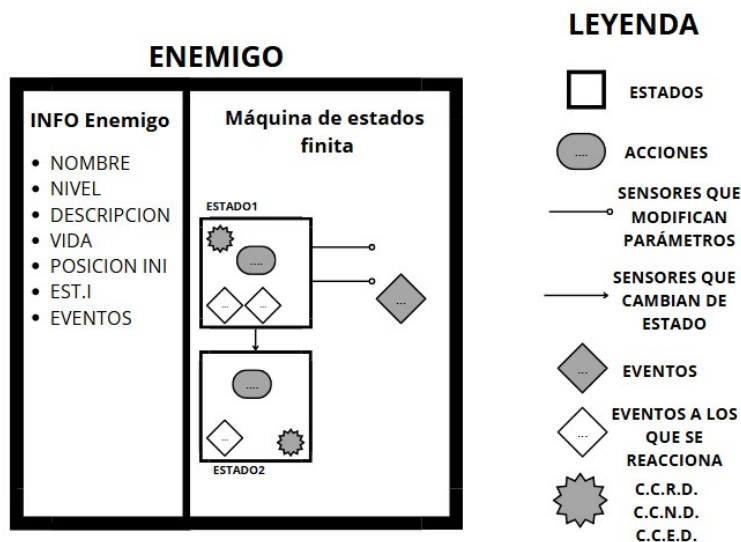


Figura 3.1: Enemigo General

3.3. Ejemplos

Aquí comienza la descripción del trabajo realizado. Se deben incluir tantos capítulos como sea necesario para describir de la manera más completa posible el trabajo que se ha llevado a cabo. Como muestra la figura 3.2, está todo por hacer.

Si te sirve de utilidad, puedes incluir tablas para mostrar resultados, tal como se ve en la tabla 3.1.



Figura 3.2: Ejemplo de imagen

Col 1	Col 2	Col 3
3	3.01	3.50
6	2.12	4.40
1	3.79	5.00
2	4.88	5.30
4	3.50	2.90
5	7.40	4.70

Tabla 3.1: Tabla de ejemplo

Capítulo 4

Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de grado, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF.

Introduction

Introduction to the subject area. This chapter contains the translation of Chapter 1.

Conclusions and Future Work

Conclusions and future lines of work. This chapter contains the translation of Chapter 4.

Contribuciones Personales

En caso de trabajos no unipersonales, cada participante indicará en la memoria su contribución al proyecto con una extensión de al menos dos páginas por cada uno de los participantes.

En caso de trabajo unipersonal, elimina esta página en el fichero `TFGTeXiS.tex` (comenta o borra la línea `\include{Capitulos/ContribucionesPersonales}`).

Estudiante 1

Al menos dos páginas con las contribuciones del estudiante 1.

Estudiante 2

Al menos dos páginas con las contribuciones del estudiante 2. En caso de que haya más estudiantes, copia y pega una de estas secciones.

Apéndice **A**

Título del Apéndice A

Los apéndices son secciones al final del documento en las que se agrega texto con el objetivo de ampliar los contenidos del documento principal.

Apéndice	B
----------	----------

Título del Apéndice B

Se pueden añadir los apéndices que se consideren oportunos.

Este texto se puede encontrar en el fichero Cascaras/fin.tex. Si deseas eliminarlo, basta con comentar la línea correspondiente al final del fichero TFGTeXiS.tex.

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

*–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

