Resumen

**Obsérvese que esta página es la página i y las secciones siguientes, las previas al primer capítulo de la memoria se numeran en números romanos en minúsculas.**

**En este apartado se incluye el resumen del proyecto (de 200 a 250 palabras), que debe describir el proyecto realizado, es decir, se basará en la descripción del proyecto que se presenta en la solicitud de registro del mismo, pero en dicha descripción el proyecto aún no estaba realizado, y en la entrega del mismo sí, por lo que es necesario revisar al menos los tiempos verbales, además de aportar más detalle, al estar el proyecto terminado. En este resumen debería incluirse el área de conocimiento principal, una breve descripción de las tareas de que consta el proyecto (estudio, planificación, mejora, desarrollo…), y el alcance específico del mismo (sobre todo si se trata de un subproyecto dentro de un proyecto más grande). Resumen y descriptores no deben superar la página de longitud, y esta página lleva el mismo encabezado que el resto del documento.**

Este Trabajo de Fin de Grado, consiste en el desarrollo de un videojuego de investigación criminal utilizando el motor Unity 3D, con posibilidad de ejecución en cualquier ordenador mediante un ejecutable. El objetivo principal es el de la generación procedural de crímenes, de manera que cada partida presente un caso único, con variaciones en el tipo de crimen, entorno, sospechosos, pruebas y eventos.

Para lograr este objetivo, se ha diseñado y desarrollado un sistema de interrogatorios dinámico, en el que los personajes no jugables (PNJs) responderán en tiempo real a las preguntas del jugador. Para ello, se ha implementado un modelo de inteligencia artificial que permitirá generar respuestas coherentes basadas en la información del caso y el comportamiento del sospechoso, siendo este personalizado para cada PNJ y distinto en cada ejecución.

El propósito es crear una experiencia lo más inmersiva posible, en la que el jugador deba analizar pistas, interrogar a testigos y sospechosos, y conectar la información recopilada para resolver el crimen. La generación procedural permitirá que cada partida sea diferente, mientras que el sistema de IA aportará mayor profundidad a las interacciones con los personajes.

A nivel técnico, él se han implementado una serie de algoritmos de generación procedural para los casos, un motor de diálogo interactivo y un sistema de toma de decisiones basado en IA.

Descriptores

**Aquí deben aparecer entre 3 y 5 palabras clave (descriptores) que ayuden a clasificar adecuadamente el proyecto.**

Videojuego, Unity, Investigación, Generación procedural, Inteligencia Artificial

Índice

Tabla de contenido

[1. INTRODUCCIÓN 1](#_Toc197692777)

[2. JUSTIFICACIÓN Y ANTECEDENTES 3](#_Toc197692778)

[2.1 JUSTIFICACIÓN 3](#_Toc197692779)

[2.2 ANTECEDENTES 4](#_Toc197692780)

[2.2.1 Shadows of Doubt 4](#_Toc197692781)

[2.2.2 Suck Up! 5](#_Toc197692782)

[3. OBJETIVOS Y ALCANCE DEL PROYECTO 6](#_Toc197692783)

[3.1 OBJETIVOS PRINCIPALES 6](#_Toc197692784)

[3.2 OBJETIVOS SECUNDARIOS 6](#_Toc197692785)

[3.3 objetivos fuera del alcance 7](#_Toc197692786)

[4. ESTUDIOS DE TECNOLOGÍAS A UTILIZAR 8](#_Toc197692787)

[4.1 Tecnologías software 8](#_Toc197692788)

[4.1.1 Motor de Videojuegos 8](#_Toc197692789)

[4.1.2 Herramienta de gestión de versiones 9](#_Toc197692790)

[4.1.3 Software de edición de texto 9](#_Toc197692791)

[4.1.4 Software de gestión de planificación 10](#_Toc197692792)

[4.2 TECNOLOGÍAS INVESTIGADAS 10](#_Toc197692793)

[4.2.1 Generación procedural 11](#_Toc197692794)

[4.2.2 PNJs con Inteligencia Artificial 16](#_Toc197692795)

[5. PLANIFICACIÓN Y ANÁLISIS DEL PRYECTO 20](#_Toc197692796)

[5.1 PLANIFICACIÓN DE TAREAS 20](#_Toc197692797)

[6. DISEÑO DE LA SOLUCIÓN 21](#_Toc197692798)

[7. DESARROLLO DE LA SOLUCIÓN 21](#_Toc197692799)

[8. Valoración ética del proyecto 21](#_Toc197692800)

[9. CONCLUSIÓN Y REFLEXIONES FINALES 21](#_Toc197692801)

[10. Bibliografía 21](#_Toc197692802)

**Tabla de figuras**

[Figura 1 Shadws of Doubt 5](#_Toc197692803)

[Figura 2 Suck Up! 5](#_Toc197692804)

[Figura 3 Logotipo del motor Unity 8](#_Toc197692805)

[Figura 4 Logotipo de Github 9](#_Toc197692806)

[Figura 5 Logotipo de Microsoft Word 9](#_Toc197692807)

[Figura 6 Logotipo de Trello 10](#_Toc197692808)

[Figura 7 Ejemplo grafo escenario 13](#_Toc197692809)

[Figura 8 A Procedurally Generated Murder Mystery interfaz web 15](#_Toc197692810)

[Figura 9 Herramienta Convai para PNJs con IA 18](#_Toc197692811)

[Figura 10 LLM for Unity 19](#_Toc197692812)

# INTRODUCCIÓN

En esta memoria se recogerá toda la información relativa a la planificación, diseño y desarrollo del proyecto de fin de grado que he realizado como paso final de mi aprendizaje en la titulación Ingeniería Informática + Videojuegos, Realidad Virtual y Realidad Aumentada, aplicando en él todas las competencias que durante estos cuatro años he ido adquiriendo en la Universidad de Deusto.

Para facilitar la búsqueda y lectura de esta información y de datos concretos, el documento se encuentra dividido en diversos capítulos. Estos capítulos se han definido en base a los apartados considerados obligatorias según las directrices y normativas de trabajos de fin de grado de la Universidad de Deusto, salvo ciertas mínimas modificaciones que han podido ser realizadas en el renombre de ciertos títulos, división de algunos apartados en ciertos capítulos o la adición o modificación de algún capítulo en específico si se considera oportuno.

Teniendo esto en cuenta, y excluyendo el propio capitulo introductorio actual, estos serían los capítulos en los que se dividirá la memoria:

* **Justificación y antecedentes**: En este capítulo se recogerá la información referente a la motivación que me ha llevado a escoger y realizar este proyecto, además del nombramiento de algunos ejemplos previos que me han servido como inspiración o modelo para el proyecto.
* **Objetivos y alcance del proyecto**: En este capítulo se definirán las distintas metas y objetivos que se plantean cumplimentar de cara al desarrollo del proyecto, además de ciertos objetivos que tuvieron que ser descartados y/o abandonados.
* **Estudios de tecnologías a utilizar**: Dado el alcance de mi proyecto y la idea de uso de ciertas tecnologías y algoritmos que no había utilizado anteriormente, utilizaré este capítulo para especificar las tecnologías utilizadas durante el desarrollo del proyecto, tanto software como hardware, además de presentar alternativas a estas y explicar los motivos por los que se han utilizado algunas y descartado otras.
* **Planificación y análisis del proyecto**: A lo largo del desarrollo del proyecto se ha llevado a cabo un plan de trabajo en el que se han establecido una serie de tareas para asegurar el cumplimiento de los objetivos definidos para el proyecto y para garantizar una organización y orden durante el desarrollo. Además, se hace un análisis de los perfiles necesarios para el correcto desarrollo de proyecto y el presupuesto requerido para el proyecto.
* **Diseño de la solución**: En este apartado se detallarán todas las ideas que se han tenido para abordar el diseño del proyecto. Se explicarán todas las decisiones de diseño, tanto como las mecánicas del juego y sus interfaces, como la parte más “artística”.
* **Desarrollo de la solución:** Este capítulo se centrará en la implementación práctica del diseño que se ha conceptualizado en los capítulos previos. Se explicarán en detalle todos los procesos de desarrollo del videojuego, incluyendo la programación de las mecánicas, implementación de tecnologías, creación de interfaces, organización de clases y todo lo referente al desarrollo del producto. También se detallarán los requisitos funcionales y no funcionales del proyecto, junto con el plan de pruebas y el manual del usuario.
* **Valoración ética del proyecto:** Este capítulo está destinado a la explicación de la valoración ética del proyecto, haciendo especial énfasis en los principios éticos aplicados y aquellos que podrían ser vulnerados.
* **Conclusión y reflexiones finales:** En este apartado, se realizará una conclusión final del proyecto, explicando los desafíos enfrentados a lo largo de todo el proceso, analizando los resultados obtenidos y valorando como este trabajo podría haberse mejorado de cara al futuro. Se propondrán además posibles mejoras y ampliaciones futuras, además de evaluar en qué medida los objetivos del proyecto han sido cumplidos.
* **Bibliografía:** Por último, se utilizará este capítulo final para reunir todas las fuentes de información usadas para la elaboración de este proyecto de fin de grado, tanto en el momento de la planificación, como en el de desarrollo de proyecto o en la propia realización de esta memoria.

# JUSTIFICACIÓN Y ANTECEDENTES

El objetivo de este apartado será el de justificar los motivos que me han llevado a la elección y diseño de un videojuego/simulación de investigación de crímenes generados de forma procedural con Inteligencia Artificial integrada como proyecto de fin de grado, además de mencionar los juegos que me han servido de una forma u otra como inspiración para la idea o desarrollo de este.

## JUSTIFICACIÓN

La idea de desarrollar esta idea de proyecto nace de la mezcla de dos de mis mayores pasiones desde hace muchos años: los videojuegos, que me han acompañado gran parte de mi vida y me han hecho legar a estudiar una carrera como la que estoy haciendo en estos momentos; y los crímenes, temática que siempre me ha gustado desde muy pequeño y de la que siempre me empapado cuanto he podido, desde novelas negras hasta documentales de true crimes.

Inicialmente, cuando fui a realizar la idea de mi proyecto de fin de grado el año pasado la idea que fui a desarrollar era radicalmente distinta. La idea que planteé y comencé a desarrollar fue la de un videojuego en Realidad Virtual del juego de la escoba, con unos añadidos de mecánicas de juego de acción con movimientos. La idea me gustaba, pero comencé a estar muy desmotivado, y estaba pasando por un momento bastante malo, sumado al hecho de que iba a tener que ampliar mi estancia en la carrera por un año más. Es por eso que un día planteándome que hacer, me di cuenta que necesitaba que el proyecto que fuera realizar y al que le fuera a dedicar tantísimas horas tendría que ser algo que me motivase de verdad, no un simple proyecto a realizar.

Es por ello que, tras mucho pensarlo y debatirlo con personas que me conocían decidí unificar las que, como he dicho, eran mis grandes pasiones. Siendo Criminalística la carrera que siempre quise estudiar, pero a la que nunca me atreví a entrar; y siendo los videojuegos un medio de entretenimiento que desde pequeño siempre me ha acompañado, un videojuego con temática de resolución de crímenes era la idea idónea.

Una vez tenía muy en claro que la temática que quería desarrollar era la de resolución de crímenes, se me planteó la duda de que tipo de proyecto quería hacer, a la altura de un proyecto de fin de grado, y en el que aplicase gran parte de los conocimientos y competencias aprendidos durante la carrera. Fue ahí cuando, además inspirado en un videojuego ya existente llamado *Shadows of Doubt* (del que hablaré más adelante), tomé la decisión de que desarrollaría un videojuego de investigación de crímenes en el que, en lugar de estar todo scripteado como un guion, sería diferente en cada ejecución mediante generación procedural; además de el posterior añadido de la mecánica de Inteligencia Artificial a los PNJs a la hora de realizar los interrogatorios.

## ANTECEDENTES

A la hora de buscar inspiración y contextualizar el desarrollo de este PFG, es necesario echar un vistazo atrás y analizar los antecedentes en el ámbito de los videojuegos con temáticas similares. Los distintos aspectos que eran necesarios analizar para poder hacerme una idea del tipo de producto que iba a querer y sobre todo poder desarrollar. En primer lugar, se encontraban los juegos con generación procedural de cualquier tipo, para analizar qué tipo de algoritmos se utilizaban y que enfoque se les daba; y, sobre todo, los videojuegos de estilo “detectivesco” de investigación de crímenes.

A continuación, analizaré los principales juegos que se utilizaron como referencia, en mayor o menor medida, y sus contribuciones tanto a mi propio proyecto como a la industria de videojuegos en sí.

### Shadows of Doubt

En primer lugar, y siendo el videojuego mas influyente para mi PFG, ya que fue del que agarré la idea principal cuya receta traté de mejorar o al menos acomplejar con la integración de personajes con inteligencia artificial. *Shadows of Doubt* es un videojuego independiente de investigación de crímenes en primera persona desarrollado por ColePowered Games y lanzado en acceso anticipado en 2023. Con su ambientación futurística estilo noir creada completamente mediante voxels, se ambienta en una ciudad generada completamente de forma procedural, lo que ofrece una experiencia distinta en cada partida.

El principal aspecto de este juego y uno de los mas innovadores es su motor de simulación procedural, que genera una ciudad completa con sus edificios, ciudadanos con rutinas propias y una red de relaciones entre cada personaje, cada cual, con su propio nombre, ocupación domicilio, hábitos, conexiones sociales, etc. Además, los crímenes se generan dinámicamente dentro de este sistema, lo cual obliga al jugador a usar el razonamiento deductivo y la observación para resolverlos.

Desde el punto de vista jugable, *Shadows of Doubt* se caracteriza por ofrecer total libertad al jugador. No existen rutas predefinidas ni soluciones únicas para cada caso; en su lugar, el juego recompensa la creatividad, la recolección de pruebas (el arma del crimen, huellas dactilares o grabaciones de cámaras) y la conexión lógica entre pistas. Este tipo de estructura plantea un desafío técnico enorme, ya que requiere que los sistemas de generación y los mecanismos del juego estén medidos al milímetro para que todo tenga una conexión lógica.

En resumen, este juego no solo me ha servido como referencia para definir el tipo de experiencia que quería desarrollar, sino también como un estudio de caso sobre cómo puede llegar este juego a implementar crímenes de forma procedural en un entorno simulado y como poder replicarlo.



Figura Shadws of Doubt

### Suck Up!

*Suck Up!* es un videojuego independiente que destaca en la integración de PNJs con IA, lo cual lo hace muy relevante para tener en cuenta a la hora de investigar mi proyecto. En este título, el jugador encarnará a un vampiro que deberá infiltrarse en distintos hogares de un vecindario, para lograr alimentarse sin ser detectado. La parte técnica innovadora de este título son las reacciones dinámicas de los personajes a la presencia y acciones del jugador. Los personajes pueden reaccionar a comportamientos sospechosos por parte del jugador, modificando su comportamiento en consecuencia. Además, parte del gameplay consiste en hablar con estos PNJs tratando de convencerles de que te dejen entrar en sus hogares, siendo la IA que tienen integrada la que valorará si les has convencido o no, además de darte respuestas acordes a la situación. Este título ha servido como referencia para poder crear personajes que no tienen diálogos y acciones predefinidas, sino que son capaces de adaptar su comportamiento según distintos estímulos, algo crucial para generar crímenes creíbles y resolverlos mediante observación y deducción.



Figura Suck Up!

# OBJETIVOS Y ALCANCE DEL PROYECTO

El objetivo principal de este proyecto es el diseño y desarrollo de un videojuego de investigación de crímenes generados de forma procedural en primera persona. Para lograr enfocar este objetivo de una manera más realista y sencilla he desglosado este objetivo principal en distintos objetivos más específicos que podrán servir además como metas que alcanzar durante el desarrollo, además de servir para comprobar si el proyecto se desarrolla exitosamente.

## OBJETIVOS PRINCIPALES

En este primer apartado definiré cuáles son los objetivos troncales del proyecto, y por tanto, aquellos que deberán ser añadidos obligatoriamente al proyecto para poder determinar si se ha alcanzado la meta principal que había con este proyecto.

1. Desarrollo de jugabilidad en primera persona: Se deberá garantizar que la experiencia se presenta desde una perspectiva inmersiva en primera persona, incluyendo además la posible interacción con PNJs y objetos del entorno.
2. Implementación de un algoritmo de generación de crímenes procedural: El sistema deberá ser capaz de crear distintos escenarios de crímenes variados, definiendo cada vez una víctima, un culpable, arma, lugar del crimen, móvil, etc. de forma coherente y dinámica.
3. Desarrollo de una simulación funcional del entorno: Deberá existir un escenario en el que los personajes, objetos y pistas se instancien de forma procedural en las distintas zonas acorde al escenario del crimen definido.
4. Integrar PNJs con inteligencia artificial: Los personajes que se instancien en el entorno deberán tener un modelo de lenguaje con el que el jugador pueda interactuar para garantizar los interrogatorios dinámicos.
5. Diseño de sistema de recopilación y gestión de pruebas: El jugador deberá poder localizar y analizar distintas pruebas físicas (arma del crimen, huellas, etc.) para avanzar en la investigación.
6. Crear un sistema de razonamiento deductivo y resolución del caso: El juego deberá permitir al jugador resolver los crímenes que se generan mediante razonamiento lógico basado en las pistas y testimonios recogidos, además de permitir la resolución del caso.
7. Implementación de interfaz de usuario intuitiva: Aunque la idea del juego no plantee una interfaz de usuario muy amplia, esta deberá ser clara e intuitiva para cualquier jugador.

## OBJETIVOS SECUNDARIOS

Además de estos objetivos troncales del proyecto se definirán una serie de objetivos extras que no serán de obligado cumplimiento para asegurar que el proyecto se ha realizado con éxito, pero que podrán asegurar una mejor versión de este proyecto.

1. A
2. B

## objetivos fuera del alcance

Como con cualquier proyecto que se puede realizar, y aún más en proyectos de ingeniería, la limitación en tiempo, recursos, o incluso conocimientos en ciertos casos, hay ciertos objetivos que por una razón u otra pueden llegar a terminar quedando fuera del alcance del proyecto.

En esta sección nombraré alguno de esos objetivos que al momento de diseñar y planificar el proyecto plantee como objetivos a tratar de realizar pero que por una u otra razón han tenido que quedarse fuera.

* A
* B

# ESTUDIOS DE TECNOLOGÍAS A UTILIZAR

La elección de las herramientas, plataformas y tecnologías empleadas para el desarrollo de este juego ha sido de vital importancia para garantizar un correcto desarrollo, rendimiento y, además, para facilitar la planificación y documentación del juego.

En este apartado hablaré tanto de tecnologías software tales como: el motor de videojuegos seleccionado, herramientas de gestión de versiones, software de edición de texto, herramientas de gestión de planificación; como de tecnologías que he tenido que investigar para las mecánicas del juego tales como: tecnologías de generación procedural, tecnologías de LLM (Large Language Models).

## Tecnologías software

Comenzaré primero con este primer grupo de tecnologías. Aquellas que consisten en aplicaciones o programas de software que me han ido aportando distintas funcionalidades a lo largo del desarrollo del proyecto.

### Motor de Videojuegos

A la hora de realizar un videojuego es de vital importancia la elección o el diseño de un motor gráfico de videojuegos, ya que este va a ser tu entorno de trabajo y será una pieza fundamental del proyecto. Debido a que la idea de desarrollar un motor de videojuegos desde cero o incluso el llegar a desarrollar el proyecto sin ningún motor gráfico eran totalmente inviables y fueron descartadas inmediatamente, he optado por utilizar el motor con el que mas familiarizado me encuentro debido a su previo uso durante la titulación de *Ingeniería Informática + Videojuegos, Realidad Virtual y Aumentada.* Este motor se trata del de Unity, en su versión 2022.3.57f, una versión un tanto obsoleta, pero una de las que mas familiares me resultaba, además de ser suficiente para la realización del proyecto propuesto.

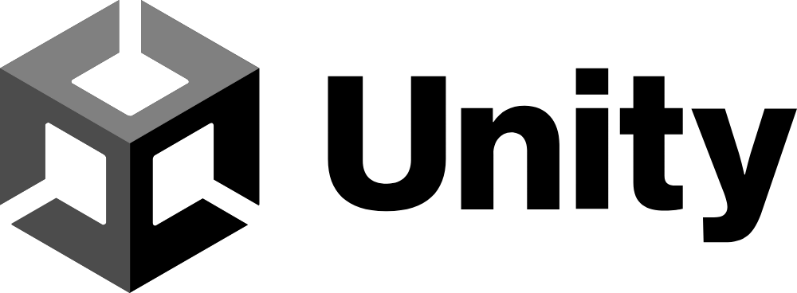


Figura Logotipo del motor Unity

Unity es una plataforma de desarrollo de videojuegos 2D y 3D ampliamente usada en todo el mundo. Además, mediante esta plataforma se pueden desarrollar juegos para muchas plataformas diferentes tales como: Windows, Linux, iOS, Android, etc. teniendo también opciones de desarrollo para plataformas VR. Desde su salida en junio de 2005 Unity no ha parado de crecer llegando a ser una de las principales y más conocida herramienta para desarrolladores, tanto independientes como grandes empresas.

Como bien había comentado antes, he elegido esta plataforma debido a que ya me sentía familiarizado con su funcionamiento además de que, debido a la gran comunidad que tiene detrás, es muy sencillo buscar ayuda o soluciones cuando tienes cualquier problema a o largo del desarrollo. Esto, sumado a la cantidad de aportaciones de la comunidad que puedes conseguir en la AssetStore de Unity la convierte en una opción genial.

### Herramienta de gestión de versiones

Para la realización de cualquier proyecto de ingeniería, sobre todo en una ingeniería informática, tanto con muchos participantes como en solitario creo que es altamente recomendable el uso de una herramienta que facilite el control sobre nuestro código, tanto para poder coordinar nuestro trabajo con posibles compañeros de trabajo como para mantener un control de versiones mientras trabajamos en el proyecto. Una de las principales y más usadas herramientas para este uso Github.



Figura Logotipo de Github

En el caso de mi proyecto he utilizado Github a través de la aplicaciónde Github Desktop, debido a que tiene una interfaz muy sencilla que ya conocía de otros proyectos previos. Hehecho uso de esta herramienta para mantener un control de versiones cuando realizaba cambios grandes en el proyecto, de forma que pudiera anticiparme a cualquier fallo, el cual podría revertir con facilidad. Además, esto me permite también mantener un control intenso acerca del flujo de desarrollo del proyecto.

### Software de edición de texto

En la etapa de realización de la memoria del proyecto, he requerido de una herramienta de creación de documentos de texto. Dentro de la gran cantidad de herramientas que hay disponibles con este propósito las principales que me planteé utilizar eran LaTeX y Microsoft Word, siendo esta última por la que decidí decantarme.



Figura Logotipo de Microsoft Word

Microsoft Word es probablemente una de las herramientas de creación y edición de documentos de texto mas utilizada en el mundo, junto con todo el conjunto de aplicaciones de Microsoft. Esta herramienta no solo ofrece una gran variedad de funcionalidades y herramientas que permiten diseñar documentos con suma facilidad sino que la gran “amplitud” de esta herramienta causa que lleve usandola durante muchos años, siendo con diferencia una de las herramientas con las que mas familiaridad siento.

Hubo un tiempo en el que me planteé utilizar LaTeX, ya que la había utilizado anteriormente y el resultado final podía haber sido muy profesional. Pero la complejidad añadida del uso de esta herramienta, sumada a mi poca familiaridad con él me hicieron terminar decantándome por Microsoft Word.

### Software de gestión de planificación

Para esta tarea hice uso de la herramienta de gestión de proyectos basada en web de Trello. Esta herramienta utiliza un sistema de tableros, listas y tarjetas para organizar proyectos y tareas. En esta herramienta cada tablera representa un proyecto distinto, dentro de estos las listas representan las distintas etapas del proyecto y, por último, las tarjetas representan las tareas individuales.



Figura Logotipo de Trello

Esta herramienta me ha sido muy útil para poder llevar un control del desarrollo del proyecto bajo un marco SCRUM. Por cada tarea que se crea se puede añadir información extra relevante sobre: que usuarios forman parte de cada tarea (para desarrollos con más personas), clase de tarea (si esta es de programación, UI, arte, etc.), fecha de inicio y fin, añadido de una checklist si la tarea es más compleja e involucra subtareas, etc.,

En un principio pensé en utilizar la opción que Github da para esta tarea ya que fue una que ya había utilizado anteriormente, tanto en la carrera como a lo largo de un proyecto personal que desarrollé en grupo con una serie de amigos, siendo esta de gran utilidad. Sin embargo, me pareció que esta herramienta se vería visualmente más clara, además de que llevaba tiempo queriendo utilizarla.

## TECNOLOGÍAS INVESTIGADAS

En este segundo grupo realizaré una explicación acerca de la investigación que realicé previa al comienzo del desarrollo del proyecto para conocer y diseñar la implementación de ciertas mecánicas. Siendo este uno de los proyectos más complejo que había realizado y debido a la idea de realizar unos hitos mecánicos de los que desconocía la forma de abordarlos completamente tuve que dedicar un tiempo antes de comenzar el proyecto en si a la investigación y planificación de estas, abordando distintas opciones.

### Generación procedural

Mientras investigaba la forma de como implementar un sistema de generación procedural de crímenes a mi proyecto pasé por distintos papers y estudios con muchos enfoques distintos, varios de los cuales descarté, aunque en general todos sirvieron para entender un poco más como terminaría haciéndolo.

#### Lógica Epistémica Dinámica (DEL) y el sistema Ostari

El sistema Ostari, desarrollado por Markus Eger y presentado en su artículo *"Ostari: Procedural Generation of Mystery Games Using Dynamic Epistemic Logic"*, implementa el modelo de Lógica Epistémica Dinámica para modelar conocimiento y creencias en los distintos personajes del entorno. Este enfoque permite tras crear el crimen representar ciertas situaciones en las que un personaje tiene sospechas de que otro personaje es el culpable.

En la práctica lo que Ostari hace, siguiendo este modelo, es lo siguiente: Define un escenario de un crimen (quién, qué, dónde) y los personajes involucrados (sospechosos, víctimas, culpables, etc.), después asigna un conocimiento inicial a cada posible personaje, siendo que cada personaje tiene cierto conocimiento parcial si es un testigo o completo si es el asesino, pudiendo cada personaje saber más, menos o estar desinformados; después Ostari utiliza DEL para actualizar el modelo epistémico cada vez que ocurre un evento (descubrimiento de pistas, acusación, interrogatorio) ajustando lo que cada personaje va creyendo.

Este sistema es muy potente ya que simula un misterio dinámico en lugar de tener un crimen estático, aportando realismo y vida al juego. Además, esto permite el uso de mecánicas más profundas, ya que puede dar a un personaje que oculta datos o que haya contradicciones realistas entre personajes.

Esta tecnología y planteamiento es muy interesante, pero a su vez es altamente compleja, incluyendo el uso de un nuevo lenguaje de programación basado en DEL y en el conocimiento de cada agente, además de que gran parte del funcionamiento y el uso de esta herramienta es posterior a la generación del crimen, que es la parte que estaba investigando. Es por eso que termine descartando esta tecnología.

#### ClueGen: Generación Procedural de Historias de Misterio

En el paper “*ClueGen: An Exploration of Procedural Storytelling in the Format of Murder Mystery Games”* publicado por Andrew Stockdale,se presenta ClueGen, una herramienta que explora la narración procedural en juegos de misterio de asesinatos. Genera narrativas completas de asesinatos, incluyendo en estas personajes, motivos y pistas, asegurando coherencia en la historia generada. Este enfoque es muy útil para crear múltiples historias únicas y coherentes en juegos de misterio.

ClueGen genera un conjunto de personajes, cada uno con atributos personales tales como nombre, ocupación, personalidad y relaciones con otros personajes. Estas relaciones (amistad, enemistad, parentesco) influyen en los motivos y comportamientos de los personajes en la narrativa.

Se selecciona aleatoriamente un personaje como víctima y otro como asesino, asegurando que exista una motivación plausible basada en las relaciones y atributos generados. Además, se determinan también el arma y lugar del crimen.

Después, ClueGen genera pistas que apuntan al asesino, así como pistas falsas (que llaman red herrings) que pueden desviar al jugador. Estas pistas se distribuyen entre los personajes y el entorno, y su credibilidad depende de factores como la personalidad del personaje que las proporciona.

Por último, el jugador interactúa con los personajes, recopilando pistas y testimonios para deducir la identidad del asesino. El sistema asegura que siempre haya suficiente información para resolver el caso, manteniendo la coherencia narrativa y la jugabilidad.

Este sistema está muy bien y es muy similar al que he terminado abordando yo en mi proyecto (aunque luego encontré uno aún más parecido a mi propuesta que se verá más tarde) aunque este tiene un enfoque mucho mayor en desarrollar una historia coherente, priorizando las relaciones entre personajes. Este artículo tiene varios apartados enteros enfocados en este apartado, aunque he omitido parte de esa información en esta memoria y durante mi investigación ya que esto se alejaba un poco del enfoque que yo quería darle al juego.

#### Generación procedural basada en simulación de líneas temporales

Uno de los enfoques más innovadores que he encontrado para la generación procedural de crímenes proviene de un prototipo compartido en un post de Reddit, en una comunidad de desarrollo de juegos. Este método no solo genera un escenario de un crimen, sino que simula activamente la evolución de este crimen en el tiempo, haciendo uso de una línea temporal y una red de personajes en un escenario específico.

El primer paso consiste en la creación de forma procedural de elementos básicos:

* El escenario, siendo este un espacio cerrado como una casa. Este se modela como un grafo de habitaciones, siendo cada habitación un nodo y los pasillos y puertas las aristas que permiten el movimiento.
* Los personajes, siendo cada uno generado con: una serie de atributos físicos y de personalidad, relaciones interpersonales y memorias individuales. Además, aquí se le asignará a un personaje el papel de asesino y se le asignará una víctima que será su objetivo.

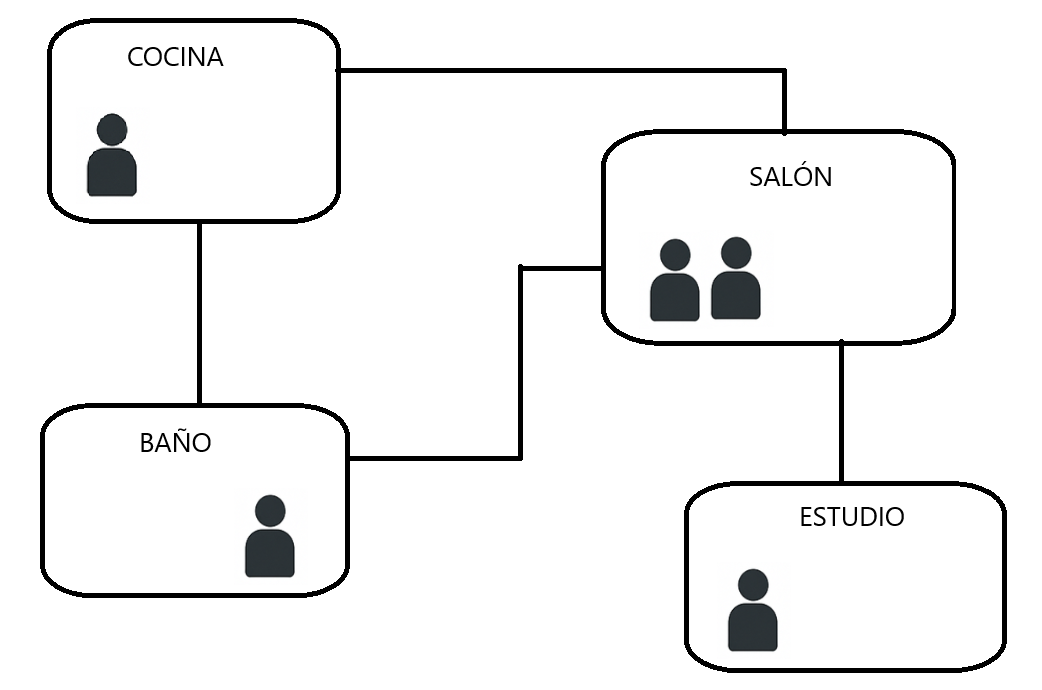


Figura Ejemplo grafo escenario

El crimen se simula después como una línea de tiempo de eventos discretos. La idea principal es la de distribuir inicialmente a todos los personajes en distintas habitaciones de forma aleatoria. Luego durante cada paso de la simulación, cada personaje podrá realizar una única acción básica, como, por ejemplo:

* Moverse de una habitación a otra.
* Hablar o bailar con alguien (deberá estar en la misma habitación que esa persona)
* Pedirle a alguien que abandone la habitación
* Permanecer quieto
* Matar a la víctima (en el caso del asesino)

La simulación avanzará paso a paso, hasta que el asesinato se haya realizado, y los personajes seguirán interactuando después, lo cual enriquece las pistas y las memorias de los personajes que actuarían como testigo.

Además, es importante el papel del asesino en este planteamiento, ya que este no ejecutará el asesinato de forma aleatoria, sino que se rige por una serie de normas o metas para cumplir este objetivo. Su meta principal será la de asesinar a su objetivo, pero tendrá submetas tales como: asegurarse de que la víctima esté en la misma habitación que él y que el lugar esté vacío de otros testigos. Para ello, el culpable trabaja hacia atrás desde esta meta principal, es decir, en caso de haber más gente en la habitación les pedirá que se vayan, además de poder moverse para seguir a la víctima o esperar a que las condiciones sean óptimas. Este tipo de enfoque, llamado planificación basada en metas (o goal graph), permite que los crímenes sigan una lógica interna y no den sensación de ser arbitrarios.

Mediante este enfoque cada personaje tiene una memoria simulada de los eventos ocurridos (Qué vieron, donde estaban, con quién hablaron, etc.). Además, podría añadirse una capa de complejidad añadiendo la opción de que estos recuerdos puedan ser distorsionados o falsos, añadiendo un nivel extra de complejidad a la deducción, ya que el jugador debería decidir en quién confiar y cómo interpretar testimonios potencialmente falsos o distorsionados.

El autor del prototipo proporcionó transcripciones de simulaciones que ilustran como se desarrolla la acción en secuencias simples, por ejemplo:

**#Case 1 ---**

**Characters: Tony, Elite, Boon (Elite wants to kill Tony)**

**-------**

**Tony dance**

**Elite ask\_to\_move -> Boon**

**Boon move**

**---------**

**Tony move**

**Elite kill Tony**

**Boon talk**

**---------**

**Elite dance**

**Boon talk**

**-----------------------------------**

**#Case 2 ---**

**Characters: Elite, Boon, Sarah, Abe, June (Elite and Sarah want to kill everyone)**

**-------**

**Elite -> kill -> Boon**

**Sarah -> kill -> June**

**Abe -> dance [does not see it happen]**

**---------**

**Elite -> dance**

**Sarah -> idle**

**Abe -> idle**

**---------**

**Elite -> dance -> Abe**

**Sarah -> dance**

**---------**

**Elite -> dance**

**Sarah -> kill -> Abe**

**---------**

**Elite -> idle**

**Sarah -> idle**

**---------**

**Elite -> talk -> Sarah**

**--------------------------------------**

En estas secuencias se puede comprobar como los asesinos manipulan la situación para quedarse a solas con las victimas antes de acabar con ellas o como se mueven para acabar en la misma habitación que ella, siguiendo un plan táctico que surge del propio sistema procedural.

En resumen, este enfoque combina generación procedural estructurada, simulación de personajes y modelado de la memoria de estos para crear crímenes ricos y dinámicos. Es un sistema muy acertado para la idea de ofrecer un misterio que sea único en cada partida y completamente deducible, aportando profundidad tanto lúdica como narrativa. Sin embargo, decidí que este enfoque se me acomplejaría si iba a hacer una simulación completa antes de comenzar lo que sería el bucle de juego de mi proyecto, además de que implementarlo junto el resto de mecánicas que quería implementar sentía que iba a hacer demasiado complejo el proyecto. Es por ello que decidí descartar este enfoque, aunque se transformó en uno de mis favoritos.

#### Generación Procedural de Misterios de Asesinatos

Encontré también un artículo llamado *"Eliminating the Impossible: A Procedurally Generated Murder Mystery",* en el que se presenta un juego de detectives web en el que los personajes y sus objetivos son generados de forma procedural. Uno de estos personajes comete un asesinato, y el jugador debe resolver el crimen recolectando evidencias e interrogando a los personajes. El sistema garantiza que el crimen siempre sea resoluble, incluso si algunos personajes intentan ocultar información.

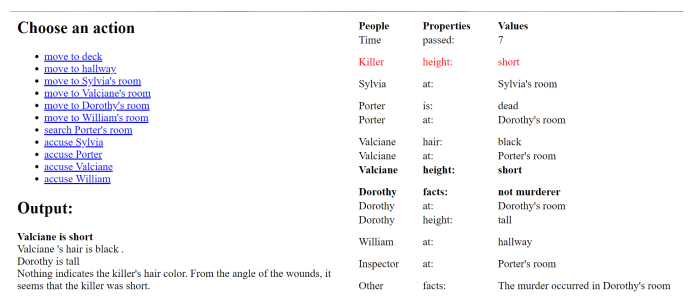


Figura 8 A Procedurally Generated Murder Mystery interfaz web

Este sistema busca aumentar la rejugabilidad de los juegos de misterio, donde normalmente una vez resuelto el caso el interés disminuye. Para ello busca la forma de generar casos únicos, asegurando que siempre exista una solución lógica accesible al jugador.

La generación procedural crea desde cero un escenario criminal único mediante los siguientes pasos:

* Se generan inicialmente un conjunto de personajes (PNJs) cada uno con: nombre, relaciones entre sí (amigos, enemigos, conocidos, familia, etc.), objetivos personales (que pueden ser ajenos o no al crimen). Uno de estos personajes se asigna como asesino.
* Se define el crimen: la víctima, el arma, el lugar, etc. Esta será la resolución a la que el jugador deberá llegar.
* El sistema genera una serie de pistas físicas (huellas, sangre, objetos encontrados) y testimonios (lo que los personajes dicen que vieron o que saben). De esta forma, cada pista tiene un origen lógico, aunque podrá ser verdadera o falsa en función de quien la proporciona y con que intención.

Después de esto, al igual que en el caso de Ostari, cada personaje tiene un modelo epistémico integrado de forma que: el asesino sabe quién cometió el crimen y puede intentar ocultar esa información, los testigos tienen conocimientos parciales (por ejemplo, han podido ver a alguien cerca de la escena del crimen, pero puede ser una persona cualquiera), otros personajes pueden tener suposiciones incorrectas o desconocer datos. Cuando el jugador interactúa con estos personajes va actualizando el conocimiento de estos personajes.

Pero, si algunas pistas pueden ser confusas y además pueden mentirte en los testimonios ¿Cómo se garantiza que el crimen sea resoluble? Esto se logra mediante un sistema de generación supervisado, donde se controla la dispersión y calidad de la información, dejando siempre un “camino lógico” que seguir. De esta forma el algoritmo de generación procedural no es completamente aleatorio, sino que sigue ciertas reglas estructurales.

Imaginemos que el sistema genera:

* **Personajes**: Alice, Bob, Carol, Dave.
* **Víctima**: Carol.
* **Asesino**: Bob.
* **Arma**: Pistola.
* **Lugar**: La cocina.

Pistas generadas:

1. Huellas de Bob en la cocina.
2. Testimonio de Alice: "Vi a Bob y Carol discutiendo."
3. Testimonio de Dave: "Escuché un disparo en la cocina."
4. (Falsa) Testimonio de Bob: "Yo estaba en el jardín todo el tiempo."

El sistema asegurará en ese momento que: al menos 2-3 pistas de las generadas apunten a la persona correcta y que se añada alguna pista distractora o falsa para añadir dificultad. Además, cuando el jugador encuentra una pista física y confronte a Bob mediante un interrogatorio, la lógica epistémica actualiza el conocimiento de todos los PNJs, y Bob puede cambiar su testimonio, o Alice puede revelar más información si ve que Bob miente.

En conclusión, este sistema combina generación procedural estructurada + lógica epistémica dinámica para crear misterios que son: únicos en cada partida, complejos y realistas y, sobre todo, lógicamente resolubles.

Este fue el artículo que más me pareció que podía ir acorde al enfoque que yo quería darle a mi proyecto, por lo que fue el que más utilicé como modelo de cómo crear un algoritmo de generación del crimen, siguiendo unos pasos muy similares a los vistos en esta propuesta.

### PNJs con Inteligencia Artificial

Además de investigar algoritmos y formas de generación procedural otro aspecto troncal de mi proyecto que tenía que investigar también se trata de la implementación de Inteligencia Artificial a los personajes del juego, para poder realizar los interrogatorios con los PNJs, de forma que puedan actuar y responder de manera coherente y creíble durante la experiencia del juego.

Durante esta investigación he explorado distintas tecnologías y distintos enfoques, algunas de ellas mas como meras ideas y otras como herramientas ya accesibles para uso. A continuación, relataré aquellas que han sido más relevantes.

#### Toward Characters Who Observe, Tell, Misremember, and Lie

En la primera fase de investigación del proyecto me encontré con este artículo, publicado por un equipo de investigación de AIIDE, en el cual se propone un marco para PNJs que no solo observan los eventos que ocurren a su alrededor, sino que también tienen la capacidad de comunicar información al respecto, olvidar ciertos detalles, tener recuerdos no muy desarrollados o incluso falsos o incluso mentir de forma deliberada. Este modelo se basa en la idea de enriquecer la experiencia narrativa y la inmersión simulando comportamientos humanos como la propagación de rumores y los malentendidos.

Este enfoque se basa en ciertos componentes clave:

* La adquisición de conocimientos por parte de los PNJs, permitiendo a estos observar eventos del juego o conversar con otros personajes, haciendo que estos vayan aprendiendo a lo largo de la simulación.
* Hacer que la memoria de los personajes sea dinámica, es decir, que la memoria que tienen almacenada pueda ser degradada con el tiempo, o cambiar según ciertas influencias.
* Añadir un mecanismo de engaño, permitiendo al personaje mentir de manera deliberada, modificando su testimonio según le convenga.

Este sistema fue implementado en el juego *Talk of the Town,* un simulador social donde los personajes forman una comunidad que se siente muy viva. El mayor atractivo de este juego es como se implementa esta gestión de creencias, donde cada personaje tiene su propio modelo de conocimiento.

Este planteamiento es muy potente, pero su nivel de complejidad es muy elevado, además de estar más orientado a juegos de simulación social. Además, el enfoque que se explora en el artículo es muy teórico y hasta un poco idílico, en el sentido de no dar bases reales de cómo podría implementarse un sistema así en un proyecto real. Por ello, aunque este enfoque me sirvió para poder extraer algunas ideas de como quería que funcionase este sistema, no opté por tratar de replicar un enfoque similar a este.

#### Convai: PNJs con IA integrada

Esta es sin duda la herramienta más potente y espectacular que encontré sobre este tema. Convai es una herramienta bastante conocida usada para dotar de IA conversacional avanzada a PNJs, orientada sobre todo a videojuegos y juegos virtuales. El uso de esta plataforma permite crear personajes que: tienen un tipo de personalidad muy definida, mantienen una memoria a largo plazo acerca de sus interacciones con el jugador y además tienen la capacidad de responder al jugador en tiempo real, con texto natural y voz incluida.

Además, esta herramienta tiene una funcionalidad muy potente que a mí me pareció uno de sus puntos más llamativos, y esta se trata de la posibilidad desde la página oficial de Convai de seleccionar un modelo personalizado al gusto del usuario e implementarlo en tu proyecto con el modelo de personalidad y datos personales que se le hayan asignado. Este personaje dispondrá, además de la IA implementada, de una función de texto a voz sumada a una funcionalidad de simulación de movimiento facial y de labios al hablar. Esto le añade una capa de realismo extra a los PNJs que es muy potente y un añadido excelente.

Convai ofrece integración directa con motores como Unity o Unreal, y facilita la creación de un PNJ en pocos pasos mediante un panel web donde puedes definir su personalidad, conocimientos iniciales y reglas básicas de comportamiento.

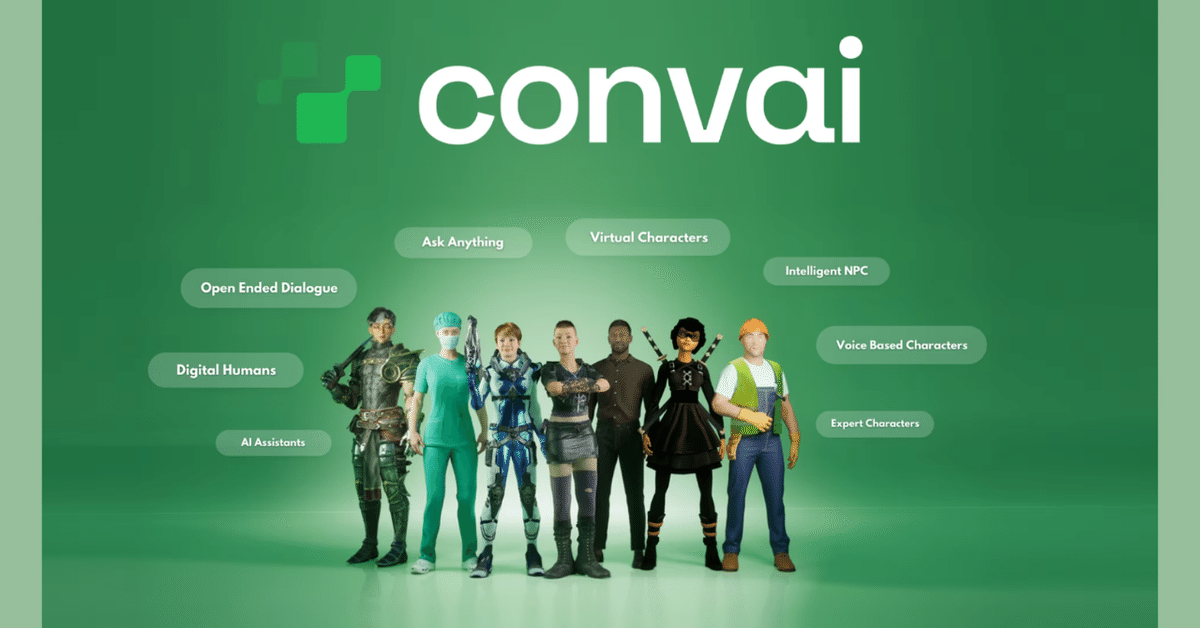


Figura Herramienta Convai para PNJs con IA

Esta herramienta es genial, y me plantee que fuera la que finalmente integraría en el proyecto, pero el problema que encontré fue que los intentos de prueba gratuitos que ofrecía esta herramienta eran demasiado limitados para el desarrollo, de forma que durante el desarrollo iba a disponer de muy pocos créditos de prueba, lo que me iba a complicar y limitar mucho, ya que a la mínima que haciendo pruebas para comprobar que la funcionalidad estaba bien implementada se iban a agotar muy rápidamente, limitando mucho el desarrollo.

#### LLM for Unity

Este asset, disponible en la Asset Store de Unity, es una integración directa de modelos de lenguaje (o LLMs) en proyectos de Unity. Este asset permite la conexión de un proyecto de Unity con un modelo GPT o similar que permite la generación de respuestas dinámicas y naturales para los PNJs en tiempo real.

Este enfoque tiene la ventaja principal de que permite la generación de diálogos no predefinidos basados en un prompt customizable por el usuario, además de facilitar una interacción mas fluida y abierta con el jugador, ya que este tiene la opción de escribir o elegir preguntas a su gusto y obtener respuestas relevantes sin depender de opciones de diálogo preseleccionadas.

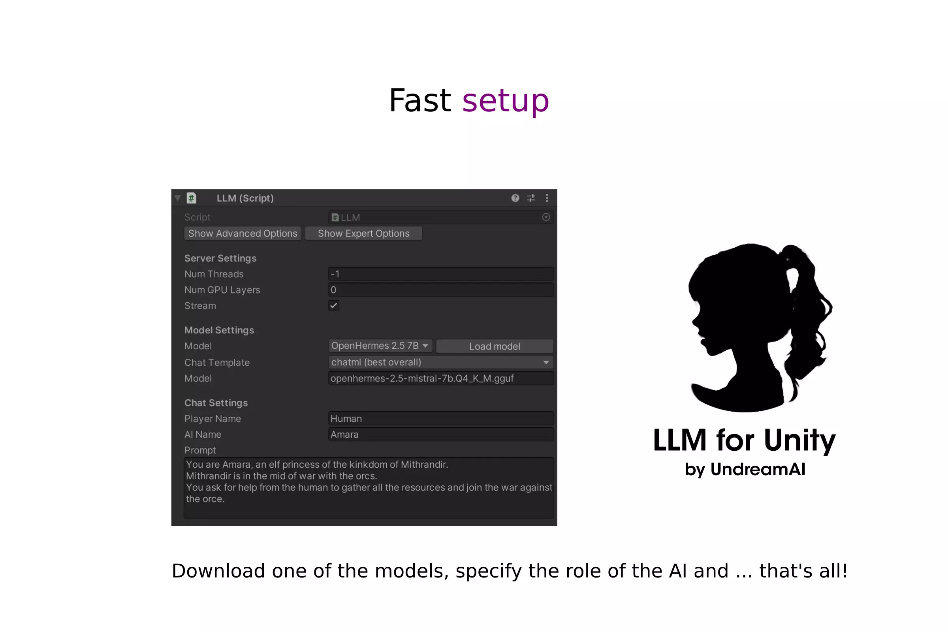


Figura LLM for Unity

Este asset ofrece además una API aparentemente sencilla (aunque a lo largo del desarrollo me haya dado algunos problemas serios que han sido complicados de arreglar) que permite el envío de mensajes y peticiones desde Unity hacia un backend de IA, recibiendo además respuestas procesadas para que el personaje las diga o que actúe en consecuencia.

Esta es la herramienta que me resultó más útil y acorde al enfoque que quería darle a mi proyecto, además de que me apareció lo suficientemente clara para que la implementación resultase lo más sencilla posible ya que, aun con las complicaciones propias del desarrollo, la herramienta esté bien documentada y trae consigo ciertos modelos y escenas prehechas que pueden ser de gran ayuda para utilizar como base para la implementación. Finalmente, y como ya he mencionado, esta fue la herramienta que terminé eligiendo para implementar en mi proyecto, después de barajar varias opciones distintas.

# PLANIFICACIÓN Y ANÁLISIS DEL PRYECTO

Para asegurar que el proyecto se desarrolla de forma óptima y que los objetivos planteados para el proyecto de fin de grado han sido cumplidos satisfactoriamente, se ha realizado un plan de trabajo que representa las distintas etapas del proyecto, las tareas planificadas para cada una de esas etapas y el tiempo empleado en cada una de estas

Además, utilizaré este capítulo para hablar de los perfiles que serían necesarios dentro de un equipo profesional de trabajo para la realización de un proyecto de este calibre, y de aquellos que serían necesarios añadir en caso de lanzar este juego al mercado, junto con un presupuesto aproximado de cuanto haría falta para este objetivo.

## PLANIFICACIÓN DE TAREAS

Para completar y lograr alcanzar todos los objetivos propuestos para el proyecto de forma óptima se ha realizado una división de tareas, que a su vez pueden llegar a dividirse en diversas subtareas, y que pueden agruparse en distintas fases o ámbitos, como podrían ser tareas de diseño, de programación, de UI, de modelado, etc.

Fase 1. Definición del proyecto y su registro

* Tarea 1. Definición inicial del proyecto. En este momento se planteó la idea general de cuál sería la temática del proyecto.
* Tarea 2. Elección y selección de tutor. Se seleccionó un tutor acorde a la idea del proyecto para presentársela y obtener un visto bueno y unas pautas para la realización de este.
* Tarea 3. Registro del proyecto. Se llevó a cabo el registro del PFG en Alud, aportando el titulo sugerido por el tutor y una breve descripción de este.

Fase 2. Planificación

* Tarea 4. Definición de objetivos. Se determinaron cuales eran aquellos objetivos principales que el juego debía tener para darlo por completado. Además, se plantearon también unos pocos objetivos de carácter más secundario, aunque a lo largo del desarrollo irían apareciendo más.
* Tarea 5. Estudio de trabajo. Se realizó un estudio para conocer el alcance de este proyecto y el enfoque que se le daría. En esta tarea se juntan tanto el estudio de otras alternativas ya existentes en el mercado que podrían servir como inspiración, como el estudio de tecnologías e ideas que podrían ser utilizadas para ver cómo se aplicarían al proyecto.
* Tarea 7. Elaboración de un plan de trabajo. Tras finalizar el estudio previo al desarrollo se marcan las directrices y tareas que se seguirán a lo largo del desarrollo, garantizando cierto orden a lo largo de este.

Fase 3. Diseño

* Tarea 8. Definición del prototipo. Se ideó la propuesta del proyecto y se planificó como quiero que sea el producto final. En esta tarea se diseñaron también los sistemas de movimiento y las mecánicas principales.
* Tarea 9. Diseño y búsqueda de assets. Se buscaron modelos 3D acorde a aquellos que se planificaron que iban a ser necesarios, como por ejemplo el escenario, los prefabs de las pistas, muebles para decorar el escenario, etc.

Fase 4. Desarrollo

* Tarea 10. Creación del esqueleto del proyecto. Se programaron todos los scripts de las principales clases que conformarían el proyecto.
* Tarea 11. Implementación de modelos en Unity. Se creó una escena en la que se añadieron todos los props y los assets que se habían seleccionado previamente. Además, al añadir el escenario se programó el script de control del jugador en primera persona, configurándolo para que el escenario fuese transitable.
* Tarea 12. Generación de crimen de forma aleatoria simple. Se programó un primer algoritmo simple en el que se generaba un escenario de crimen de forma aleatoria y se devolvía en formato JSON. De esta forma se confirmó que el algoritmo era funcional.
* Tarea 13. Creación de entorno de prueba. Para facilitar los siguientes pasos del desarrollo se creó un entorno de prueba separado del escenario principal para poder testear las siguientes funcionalidades.
* Tarea 14. Implementación de chatbot a un personaje con trazas de personalidad. Se implementó en el entorno de prueba un personaje con un chatbot instanciado para testear el funcionamiento del modelo de LLM seleccionado.
* Tarea 15. Implementación de spawneo de personajes y victimas en el entorno. Se programó la funcionalidad de que los personajes se instanciasen en el escenario (al principio esto se realizó en el entorno de prueba y posteriormente fue añadido al escenario principal) y que la victima apareciese en el suelo de la sala del crimen. Además
* Tarea 16. Implementación de mecánica de scaneo. Se implementó una mecánica de scaneo, que podía ser utilizada para recibir información acerca de los personajes y posteriormente de las armas o las pistas en el entorno.
* Tarea 17. Spawneo del arma del crimen. Se programó que el arma del crimen se instanciase en el escenario.
* Tarea 18. Spawneo de pistas. Se programó que se instanciasen en el escenario las pistas que se generan en el crimen. Entre estas se encuentran manchas de sangre en ciertos crímenes, huellas dactilares o de zapato, etc.
* Tarea 19. Adición de chatbots independientes a los personajes. Se programó que los personajes instanciados en la escena tuviesen un chatbot funcional con un prompt personalizado para cada personaje.
* Tarea 20. Resolución del caso. Se programó la mecánica que permite la resolución del caso y la comprobación de que este se ha resuelto correctamente.

Fase 5. Documentación, pruebas y optimización

* Tarea 21. Pruebas y resolución de bugs. Aunque esta tarea se fue repitiendo a lo largo de todo el desarrollo se realizó un repaso de todo lo desarrollado en búsqueda de posibles problemas para poder solucionarlos.
* Tarea 22. Refactorización y optimización. Se revisó el código en búsqueda de posibles mejoras para el rendimiento y la eficiencia de este, además de eliminar código que no se utilizase o redundancias.
* Tarea 23. Documentación del proyecto. Esta tarea consiste en la escritura de la propia memoria del proyecto de fin de grado, explicando en esta detalladamente todas las fases del proyecto y el trabajo realizado en estas, además de todo lo relevante respecto a este.
* Tarea 24. Revisión de memoria con el tutor. Se realizó la revisión final del proyecto y la memoria con el tutor, marcando y realizando los cambios necesarios.

Fase 6. Final del proyecto.

* Tarea 25. Entrega de la memoria. Se entregó la memoria del proyecto de fin de grado tras la revisión de esta mediante la plataforma Alud.
* Tarea 26. Preparación y defensa del proyecto. Se preparó y realizó la defensa del proyecto de fin de grado frente al jurado.

A continuación, he realizado una tabla que resume las distintas tareas que aquí he planteado junto a una aproximación de cuantas horas fueron necesarias para la realización de cada una.

**[ TABLA]**

Además, he realizado un diagrama de Gantt que muestra también de forma mas visual estas fases del proyecto.

**[ DIAGRAMA DE GANTT ]**

## EQUIPO DE TRABAJO

En el caso de querer realizar un proyecto de estas magnitudes de forma exitosa para, posteriormente, lanzarlo al mercado sería primordial contar con un equipo de profesionales con experiencia en los distintos ámbitos del proyecto. Todas las fases del proyecto, desde el diseño inicial hasta la creación de elementos artísticos tales como texturas o modelos 3D requieren de habilidades técnicas que una única persona no podría cubrir por sí misma durante todo el período de desarrollo.

Por mi parte he realizado ciertas tareas que son indispensables para tener un producto satisfactorio, pero el resultado no alcanza el nivel que podría tener un desarrollo con un equipo profesional desarrollado durante más tiempo con el objetivo de publicarlo. En caso de querer realizar este proyecto de mayor magnitud serían necesarios ciertos perfiles tales como:

* Diseñador de juego: Este perfil es el pilar central de todo desarrollo de un juego. Esta es la persona encargada de idear como será el producto que va a desarrollarse, y transformar esa idea en un plan de trabajo bien definido al que el resto de partes del equipo puedan ceñirse. Esta persona es la que documenta de forma muy cuidadosa todos los aspectos del proyecto, desde las mecánicas bases del proyecto hasta la forma de interacción del jugador, los personajes o incluso la narrativa.
* Programador: Otro de los pilares de un proyecto de software es la figura del programador, encargado de desarrollar el código que implementará la lógica necesaria para llevar a la vida las ideas del diseñador. El trabajo del programador no es únicamente el de codificar, sino que también el de optimizar ese mismo código asegurando una experiencia lo más óptima posible.
* Director de proyecto: Este es un perfil muy necesario a la hora de gestionar grandes equipos para un proyecto, ya que esta es la figura que supervisa y coordina a los distintos perfiles durante todas las etapas del proyecto. Durante todo el desarrollo este perfil se encargará de coordinar a todo el equipo para que las tareas y metas se cumplan y que se hagan en los plazos y tiempos previstos.
* Artista conceptual: Este perfil se encargaría de representar de una manera mucho mas visual las ideas del diseñador del juego, de forma que el resto del equipo tenga una idea mucho mas clara de cual es el objetivo a lograr.
* Modelador y animador 3D: Este perfil es muy importante de cara a la parte mas visual del proyecto, ya que partiendo de los diseños entregados por el artista conceptual creará los modelos 3D necesarios para llevarlos al proyecto.

# DISEÑO DE LA SOLUCIÓN

# DESARROLLO DE LA SOLUCIÓN

# Valoración ética del proyecto

Es importante resaltar que, desde este curso, los estudiantes debéis añadir una valoración ética de vuestro proyecto, siguiendo las directrices que habéis recibido y trabajado en la asignatura de Ética.

# CONCLUSIÓN Y REFLEXIONES FINALES

# Bibliografía

1. C. Camilli et al, “Revisión sistemática integrativa del Trabajo Fin de Grado: aspectos contextuales y pedagógicos”, Revista Complutense de Educación, 33(3) 2022: 543-553, ISSNe: 1988-2793.
2. Formato cita web: “Titulo/Nombre de la Web”, http://www.google.es, (consultado el 7/10/2022).
3. “Real Automóvil Club de España”. http://www.race.es/corporativo/noticias/-/blogs/los-accidentes-de-trafico-son-la-primera-causa-de-mortalidad-entre-los-15-y-29-anos (consultado el 7/10/2022).
4. Formato cita libro: Autores, “Título de la contribución/Libro”, *Editorial en cursiva*, Año de la publicación.
5. Formato artículo en revista/congreso: Autores, “Título del artículo”, *Nombre de la revista/congreso en cursiva*, rango de páginas, año de la publicación. (dirección web si es accesible).
6. K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, “On the ability of the 802.11p mac method and stdma to support real-time vehicle-to vehicle communication,” *EURASIP Journal on Wireless Communications and Networking - Special issue on wireless access in vehicular environments*, no. 5, January 2009.
7. Luis Dávila Gómez, Luis Castedo Cepeda, Cecilia García Cena, Cristobal Tapia Garcí, “Sistema didáctico de bajo coste para el aprendizaje de dispositivos lógicos programables”, *TAEE2012*, 105-109, 2012.

<http://taee.euitt.upm.es/actas/2012/papers/2012S3A5.pdf>

NOTA: la bibliografía deberá referenciarse a lo largo de la memoria cuando corresponda, no ser solamente un listado de fuentes consultadas sin saber para qué. Así, su numeración no sigue un orden alfabético, sino de aparición a lo largo de la memoria.

Ejemplo de referencia en el texto de la memoria:

El proyecto realizado comparte características con otros existentes **[6]**, si bien se han añadido…