

IDENTIFICACIÓN Y PROCESADO DE NARRATIVAS SECUNDARIAS EN VIDEOJUEGOS

IDENTIFICATION AND PROCESSING OF
SIDE NARRATIVES IN VIDEOGAMES



Alejandro Marín Pérez (GDV)
Carlos María González Martínez (GDV)
Carlos Jiménez Álvarez (GIS)
Alejandro Ruiz Martín (GIS)

Profesor director:
Prof. Dr. D. Carlos León Aznar

2020/2021

Trabajo de fin de grado
Grado en Desarrollo de Videojuegos
Grado en Ingeniería de Software
Facultad de Informática, Universidad Complutense de Madrid

Agradecimientos

Queremos agradecerle a Carlos, nuestro director de proyecto, la inmensa paciencia y la absoluta disponibilidad demostrada a lo largo de todo el año. Incluso ante la falta de sueño, nos ha brindado una ayuda indispensable a la hora de realizar este trabajo.

Agradecemos también a todos los encuestados, que sacrificando su tiempo libre nos han permitido llevar a cabo esta investigación. Ya hayan dedicado cinco minutos o una hora, su frustración por perderse en el mapa y su confusión por los relojes de arena estará siempre en nuestros recuerdos.

Muchas gracias a todos aquellos que han tenido que soportar nuestro estrés y nuestras reuniones, además de fregar cuando no les tocaba.

Tabla de Contenidos

Resumen	7
Palabras clave	7
Capítulo 1: Introducción	9
1.1 Motivación	14
1.2 Hipótesis de partida	14
1.3 Objetivos	15
1.4 Metodología y plan de trabajo	16
1.4.1 Fases del proyecto	17
1.5 Estructura del resto del documento	19
Capítulo 2: Estudio del trabajo previo	21
2.1 Sistemas de procesado automático de narrativa	21
2.1.1 Sistemas de gestión de la narrativa	21
Novel Writer	26
Talespin	27
Author	27
Universe	28
Minstrel	28
Agent Stories	29
Mexica	30
The Virtual Storyteller	30
Fabulist	31
Narrador en nn	31
2.1.2 Narrativa en los videojuegos	31
2.2 Tecnologías usadas en el trabajo	36
2.2.1 Herramientas de diseño de historias	37
Bibisco y Scrivener	37
2.2.2 Herramientas de planificación y control	38
Trello	38
Hojas de cálculo de Google	39
GitHub	39
2.2.3 Herramientas de redacción de la memoria y encuestas	39
Google Docs	39
Google Forms	39
Zotero	40
2.2.4 Herramientas de desarrollo	40
Unity	40
Visual Studio 2019 Community	40
Visual Studio Code	40

2.2.5 Librerías externas	41
QuikGraph	41
SimpleJSON.NET	41
Capítulo 3: Diseño del prototipo de juego y contexto narrativo	43
3.1 Diseño de historias para el análisis de la hipótesis	43
3.1.1 Contexto general de las historias	45
3.1.2 Desarrollo de las historias	46
Historia principal	46
Secundarias	47
Historia secundaria 1: Longitud de la historia	48
Historia secundaria 2: Cantidad de personajes involucrados	49
Historia secundaria 3: Cantidad de lugares recorridos	49
3.2 Diseño del prototipo de juego	50
3.2.1 Estilo del juego	51
3.2.2 Escenario	52
3.2.3 Sistemas del prototipo de juego	56
Sistema de gestión de misiones	57
Sistema de gestión de diálogos	57
Sistema de control e interacción	58
Capítulo 4: Diseño computacional del motor de procesamiento de historias	61
4.1 Arquitectura UML	61
4.2 Estructura inicial de la información	67
4.2.1 Estructura de los personajes	67
4.2.2 Estructura de los objetos	67
4.2.3 Estructura de los lugares y sus conexiones	68
4.2.4 Estructura de las historias	69
4.2.5 Estructura de los capítulos	69
4.2.6 Estructura de las escenas	70
4.2.7 Estructura de los diálogos asociados a los personajes	71
4.3 Primera versión del motor	74
4.3.1 Sistema de carga y almacenamiento de información	74
4.3.2 Sistema de peticiones del juego	74
4.4 Segunda versión del motor	76
4.4.1 Clasificación de personajes según su relevancia	77
4.4.2 Sistema de conexiones entre lugares	78
4.4.3 Sistema de clasificación de historias	80
4.4.4 Sistema de peticiones con Unity	82
Capítulo 5: Implementación	83
5.1 Tecnologías usadas en el desarrollo	83
5.2 Implementación inicial del prototipo de estudio	83
5.2.1 Sistema de interacciones	85
5.2.2 Sistema de diálogos	87
5.2.3 Sistema de misiones	89

5.3 Implementación inicial del motor	92
5.3.1 Sistema de lectura de ficheros	93
5.3.2 Sistema del motor	95
5.4 Cambios en la implementación tras el experimento	98
5.4.1 Cambios en el motor	99
5.4.2 Cambios en el prototipo	101
Capítulo 6: Pruebas del prototipo	103
6.1 Resultados	104
6.1.1 Historia 1	104
6.1.2 Historia 2	107
6.1.3 Historia 3	110
6.1.4 Historia 4	113
6.2 Discusión de los resultados de la experimentación	117
Capítulo 7: Aplicación de cambios tras las pruebas	119
Capítulo 8: Pruebas finales	123
8.1 Resultados	124
8.1.1 Historia 1	126
8.1.2 Historia 2	130
8.1.3 Historia 3	135
8.1.4 Historia 4	138
8.2 Discusión de los resultados	142
Capítulo 9: Discusión	145
9.1 Discusión de los resultados tras la experimentación	145
9.2 Discusión general del proyecto	149
Capítulo 10: Conclusiones y trabajo futuro	153
10.1 Conclusiones del proyecto	153
10.2 Trabajo futuro	155
Capítulo 11: Contribución individual	159
11.1 Alejandro Marín Pérez	159
11.2 Carlos María González Martínez	163
11.3 Carlos Jiménez Álvarez	166
11.4 Alejandro Ruiz Martín	170
Bibliografía	173
Anexo 1: Abstract	177
Keywords	177
Anexo 2: Introduction	179
A2.1 Motivation	183
A2.2 Starting hypothesis	184
A2.3 Objectives	184
A2.4 Methodology	185

A2.4.1 Project phases	186
A2.5 Document structure	188
Anexo 3: Conclusions and future work	191
A3.1 Project conclusions	191
A3.2 Future work	193

Resumen

La narrativa es un campo muy complejo y con mucha profundidad. A la hora de contar una historia entran en acción muchas variables y parámetros que no son visibles a primera vista, como narrativas secundarias o pequeñas desviaciones de la narrativa principal. En la literatura y la investigación actual no hay muchos trabajos que se centren en el procesado computacional de narrativas secundarias, centrándose en las narrativas principales como única narrativa que afecta a una historia. En cuanto a la generación automática de historias, no hay muchos trabajos que se centren en algo diferente a generar una trama principal cuando, muchas veces, las historias y las narrativas están compuestas por diferentes tipos de narrativas o arcos narrativos. Por tanto, no hay ni muchas maneras ni muchas teorías acerca de cómo clasificar diferentes narrativas para diferenciarlas entre sí y que puedan contribuir a la construcción de una narrativa más compleja y completa.

Por ello, mediante el estudio de diferentes técnicas narrativas, estructuración narrativa y técnicas de narrativa computacional, se ha implementado un motor computacional que es capaz de evaluar y cribar historias en principales y secundarias. Esto se ha hecho partiendo de la investigación realizada sobre narrativa computacional y narrativa clásica, con la que se han definido una serie de variables que, inicialmente, se creen determinantes para la clasificación. Con tal de probar que dichas variables funcionan y son correctas, se ha desarrollado un prototipo de juego simple con historias orientadas a comprobar la veracidad de cada una de las variables que componen la hipótesis de partida, realizando encuestas a los jugadores después de haber jugado para comprobar si realmente afectan a la detección y clasificación de narrativas. Tras el análisis de los datos obtenidos se ha conseguido ajustar dichas variables y añadir nuevas al proceso, implementando en el motor un algoritmo que ordena las historias aplicando estos parámetros.

Palabras clave

Narrativa Secundaria, Narrativa Computacional, Inteligencia Artificial, Creatividad Computacional, Videojuego

Capítulo 1: Introducción

La Creatividad Computacional es un campo del desarrollo del software que se ocupa de simular y replicar el comportamiento creativo de los seres humanos, incluyendo dentro de sus campos la generación e interpretación automática de la pintura, la música y la literatura. La Creatividad Computacional se ha aplicado, en particular, a la literatura creando diferentes herramientas que se aplican en diversos campos, como la poesía y la narrativa (Gervás, 2018).

Para ello ha sido necesario representar computacionalmente toda la información que comprenden estos campos. A toda esta área se la conoce como Narrativa Computacional, ya que es la que se encarga de la representación y procesamiento de historias literarias usando software y que a su vez tiene muchas cosas en común con la Creatividad Computacional. En ella están comprendidas las áreas que se encargan de contar historias, como son guiones de cine, novelas, obras de teatro y la generación de historias en videojuegos. En este ámbito se busca entre otras cosas crear historias sin la necesidad de la intervención de un ser humano o con alguna intervención humana, mediante algoritmos y métodos de inteligencia artificial o aprendizaje automático.

Incluso con décadas de estudio previas y multitud de sistemas y motores capaces de generar historias comprensibles y que siguen pautas típicas en narrativa, las historias generadas tienen ciertas limitaciones que hacen fácil su distinción de historias creadas por seres humanos, ya que no se pueden equiparar las capacidades de creación de un programa de software a las de una persona. La *artificialidad* de estas historias creadas por un sistema automatizado es evidente comparado con la creatividad de los seres humanos, lo que demuestra lo mucho que falta por avanzar en este ámbito.

Los sistemas de generación de narrativa existentes hasta la fecha son capaces de crear historias acotadas de cierta complejidad, diseñados para cumplir con un género como generar guiones de dramas televisivos, o incluyendo al usuario en la propia generación de la misma.

En el campo de los videojuegos, la narrativa puede ser algo más compleja si la comparamos con los guiones de cine o las novelas, ya que no sólo se dedica a contar una

historia, si no que permite a los jugadores interactuar con ella y *hacerla suya*, es decir, influir en la narrativa lo suficiente como para que parezca que se ha desarrollado según sus designios y dar cierta sensación de libertad y participación en la misma. Esto es algo que da mucho potencial a la narrativa en este ámbito ya que permite hacer partícipe e incluso responsable al jugador de sus propias acciones, llevándolo a vivir la narrativa de una forma totalmente diferente a como se vive en otro medio más tradicional de contar historias (Perry et al., 2009, p. 69).

A su vez, esto permite una visión mucho más personal de algunas historias y narrativas que se plantean en muchos videojuegos. Por ello se han desarrollado múltiples formas de presentarlas. Puesto que hay muchos géneros de videojuegos, como son *First Person Shooters (FPS, Juego de disparos en Primera Persona)*, *Roguelite/like (Juegos basados en la generación procedural y aleatoria de contenido al iniciar una nueva partida)*, *Action Role Playing Game (ARPG, Juego de Rol y Acción)*, *Point & Click (Aventuras Gráficas)*, *Role Playing Games (RPG, Juegos de Rol)*, Plataformas o Estrategia entre otros, cada uno de estos géneros tiene su forma de presentar esta narrativa y su forma de conducirla.

Por ejemplo, en los videojuegos de *Half Life*, *Half Life 2*, *Portal* y *Portal 2* (Valve, 1998, 2004, 2007, 2011, p. 2), unos juegos del género *FPS*, los protagonistas, que controla el jugador, son mudos y todos los personajes que se dirigen a ellos lo hacen como si se estuvieran dirigiendo al jugador directamente como si formara parte del videojuego, como *GLaDOS*, *Wheatly* o *G-Man*, los villanos que participan en estos juegos y con los que comunidades enteras de jugadores han conectado en sus sesiones de juego. Esto consigue que el jugador piense que es parte de esta diégesis y la inmersión sea mucho más completa, siendo que esta es en primera persona y eso acentúa mucho más todo lo que ocurre en estas aventuras.

Sin embargo, estos juegos no dejan de funcionar como si fueran pasillos por los que el jugador se mueve completando la historia que los desarrolladores han pensado. Algo que ofrecen los videojuegos es la interacción del jugador con la historia. El poder afectar de alguna manera u otra a los acontecimientos de la trama y construir una historia más personal.

Esta capacidad la muestran los juegos del estilo *RPG*, como *Fallout: New Vegas* (Bethesda, 2010) o *Persona 5* (ATLUS, 2016) donde dan la opción al jugador de elegir qué

camino tomar. En el caso de Persona 5 el jugador puede elegir entre varias opciones de diálogo con importancia variable en la historia principal, con una gestión libre del “día a día” del juego y relaciones con otros personajes, aunque suele haber una solución “correcta” entre muchas malas. Por otro lado, el caso de Fallout: New Vegas es digno de mención, ya que da la opción al jugador sobre qué hacer desde el minuto uno de juego, y muchas formas de solventar un problema de maneras y resultados diferentes.

A pesar de toda la libertad que ofrecen estos títulos, esta narrativa todavía se plantea por parte del desarrollador, que escribe una historia con multitud de ramificaciones y caminos que se pueden tomar y después se la presenta al jugador de manera que pueda elegir por dónde ir. Pero hay personas que juegan de formas totalmente distintas, intentando hacer sus historias lo más diferentes posibles de las del resto de jugadores.

En esto se refleja mucho la existencia de juegos de tipo *sandbox*, que funcionan, como su propio nombre indica, como un patio de juegos para niños. Proporcionan una serie de herramientas y mecánicas al jugador para que juegue y experimente su narrativa como le apetezca. Por ejemplo Minecraft (Mojang, 2009) no tiene una narrativa más allá de “Apareces en un mundo, sobrevive y mata a la dragona que se esconde en la dimensión del *End*”, sin embargo al permitir al jugador hacer lo que le plazca este va a generar su propia narrativa y hablará del juego contando una historia de cómo logró matar a este enemigo o a este otro y de cómo consiguió su primer diamante. A este tipo de narrativas se las conoce como *narrativas emergentes*, ya que van surgiendo a medida que el jugador juega y toma una serie de decisiones que no están planeadas desde un principio por el desarrollador.

En resumen, la generación de narrativa es una herramienta muy útil que ya se emplea, incluso en su estado actual, para enriquecer la experiencia de los jugadores y ofrecer mayor variedad. Ya existen videojuegos que utilizan estas técnicas para generar la totalidad de sus historias, tales como AI Dungeon (Latitude, 2019).

Sin embargo, para poder generar narrativas más complejas, es necesario analizar cómo funcionan dichas narrativas y de qué elementos están compuestas. De esta manera, se pueden aplicar una serie de algoritmos y cambios que mejorarían la generación de dichas narrativas y mejorarían la experiencia del jugador, ya que se podría realizar una diferenciación entre narrativas principales, más prioritarias, y secundarias, que complementan la experiencia.

Desde el punto de vista narrativo siempre se hace una distinción en cuanto a la importancia de la narrativa en distintos niveles. Comenzando por el más importante, se encuentra la *narrativa principal*, aquella que es considerada la más importante a ojos del autor y del público de la historia. Normalmente suele ser la que sirve de excusa para poner en movimiento la trama y todo lo que ella conlleva. Por ejemplo, la trama de la destrucción del anillo en El Señor de los Anillos (Tolkien et al., 1954), es considerada la trama principal de la narrativa que compone toda la novela.

Por otro lado se encuentran las *narrativas secundarias*, que son aquellas historias que complementan a la principal y ocurren, normalmente, al mismo tiempo pero que involucran personajes menos relevantes o que no son principales. Siguiendo con el ejemplo de la obra de Tolkien, toda la historia de Faramir con su padre sería considerada una narrativa secundaria, ya que aunque no afecta de manera directa a la narrativa principal en su totalidad, si que ayuda a complementar cuando Frodo se ve afectado por ella.

En cuanto a videojuegos esto se vuelve un poco más complejo porque las narrativas que se plantean como secundarias pueden afectar o no a la narrativa principal y pueden no intervenir en la sesión de juego de un jugador, mientras que en la de otro sí. Por tanto, no se pueden seguir los mismos parámetros. En videojuegos lo que se consideraría una narrativa secundaria es lo que se conocen como misiones secundarias, ya que son historias que el jugador decide jugar.

Por esta razón, conocer qué distingue las narrativas principales de las secundarias en videojuegos, y estudiar las características de las mismas, puede ayudar a implementar inteligencias artificiales orientadas a la generación de narrativas con un mayor nivel de cohesión con la diégesis del juego y que parezcan menos artificiales, ya que categorizar las historias individuales en principales y secundarias permite a los jugadores conocer cuál es la trama principal, comúnmente asignada por los propios desarrolladores y que se define como la más importante de las narrativas del juego, y las tramas secundarias, diseñadas para complementar la principal, contar historias menos importantes o simplemente dar la posibilidad al jugador de disfrutar de las mecánicas de juego.

Cuando no se realiza esta categorización, la percepción del jugador se vuelve más subjetiva, pues muchos de los componentes que conforman una historia son comunes en

principales y secundarias. Conocer la relevancia que el público general le da a dichos componentes permite a los sistemas automáticos hacer una generación más óptima y creativa de historias.

El obstáculo principal a la hora de categorizar las narrativas es el de representarlas computacionalmente. Ya que la narrativa es un concepto muy abstracto y difícil de cuantificar, es necesario dividir la narrativa en algo más manejable para un ordenador y, en extensión, para un programa que aplique técnicas de Inteligencia Artificial o Aprendizaje Automático. Por ello se analizó la narrativa de los juegos anteriormente mencionados y de algunos más mezclando diferentes géneros para ver cómo se representaba en ellos.

Centrando la atención en las narrativas planteadas en juegos de tipo *RPG*, se pudieron observar las limitaciones que tienen al plantear las narrativas como misiones que llevan al jugador desde el punto A al punto B y se planteó la idea de representar las historias computacionalmente de otra manera que permitiera al programa manejarlas de una forma más modular y extensible. Para ello se propuso la idea de representar las historias y narrativas de una forma un poco más clásica, observando la forma de escritura de ciertos autores y autoras de novelas que forman sus historias a través de capítulos que a su vez están compuestos por diferentes escenas siguiendo la siguiente estructura: Historia - Capítulos - Escenas (Blake Snyder).

Esta clasificación permite encapsular las historias en fragmentos muy pequeños y más manejables por el motor y entendibles también para una persona, ya que las historias suelen funcionar de forma lineal.

Sin embargo, las narrativas de un videojuego, aunque se pueden aplicar a cualquier medio, tienen muchas más cosas que las rodean. El término que se usa comúnmente es *world building*, esto es todos los lugares que componen el mundo en el que va a transcurrir la historia y los personajes que lo habitan, así como culturas, historia pasada, religiones, criaturas, etc. Todos estos datos influyen a la hora de crear una narrativa en un mundo, pero no se tratará en este trabajo.

1.1 Motivación

Planteado el problema que supone el organizar una narrativa para un videojuego, sería muy interesante tener alguna manera o herramienta capaz de organizarlas computacionalmente, llegando incluso a generarlas de forma dinámica a medida que el jugador va tomando decisiones. Además, es algo que en la industria de los videojuegos se va probando cada vez más en diferentes juegos y géneros. Por ejemplo, el videojuego *Hades* (People Make Games, 2020; Supergiant Games, 2018) ya incorpora un sistema que activa una serie de diálogos entre personajes en función de las acciones pasadas del jugador, dando lugar a situaciones realmente entretenidas y únicas, además de usando un sistema de aleatoriedad a la hora de elegir dichos diálogos. Si estos sistemas se pudieran apoyar en otro que ayudase a la organización automática de las narrativas o incluso asistiera en la creación de las mismas podría impulsar mucho más el desarrollo de este tipo de juegos.

Además, actualmente se cuenta con muchas herramientas y facilidades para poder crear estas herramientas pero todavía no se ha investigado lo suficiente como para llegar a usarlo en juegos que no sean experimentales.

Por tanto, tras este planteamiento, tiene sentido y podría ser muy útil crear un sistema capaz de organizar las diferentes narrativas de un videojuego para ofrecer una experiencia más dinámica a los jugadores, partiendo de narrativas ya escritas que puedan situarse en diferentes lugares del mapa con tal de presentar nuevas historias al jugador.

1.2 Hipótesis de partida

Partiendo del esquema de representación de narrativas planteado previamente, se puede observar que durante el proceso de creación de historias hay involucrados ciertos elementos relevantes que pueden dar lugar a una clasificación de las mismas. Por tanto la hipótesis que se plantea es que una historia principal es aquella cuya media de longitud de capítulos, cantidad de lugares visitados y número de personajes, es mayor que la del resto de historias que se plantean dentro de un juego.

Una vez planteada esta hipótesis de partida, se estudiará la influencia de estos valores para distinguir y clasificar las historias y así poder validar o rechazar dicha hipótesis,

pudiendo esta sufrir cambios en caso de que los resultados sean negativos. Para ello, se plantea una serie de objetivos centrados en alcanzar la validez o el rechazo de la hipótesis.

1.3 Objetivos

El objetivo principal del proyecto es crear un sistema capaz de identificar las historias secundarias dentro de la narrativa del juego. Para ello, se estudian los parámetros cuantificables (cantidad de capítulos, lugares visitados o número de personajes) que conforman las diferentes historias para determinar cuáles son los más determinantes al identificar una historia como principal o secundaria.

Una vez identificados los parámetros, se planea desarrollar un motor. Este motor leerá las historias previamente escritas siguiendo unas estructuras definidas y asignará a cada una la categoría de principal o secundaria analizando los parámetros que se hayan escogido.

Siguiendo esta línea, se han definido los siguientes objetivos:

- Estudio teórico del funcionamiento del procesado de narrativas.
- Análisis de los diferentes sistemas creados para la clasificación de historias.
- Diseño del funcionamiento del motor siguiendo la información obtenida de la investigación previa.
- Búsqueda de variables computacionales que sirvan para clasificar historias.
- Definición de la hipótesis de partida empleada para el desarrollo del motor que permitirá clasificar las historias midiendo las variables definidas en dicha hipótesis.
- Desarrollar un prototipo de juego sobre el que probar el motor.
- Crear una batería de historias secundarias de forma que cada una cumpla con una de las variables de la hipótesis de partida y una historia principal que cumpla con dicha hipótesis.
- Implementar una primera versión del motor que cargue las historias, identificadas previamente, y tenga las funcionalidades necesarias para poder ser usado por cualquier juego.
- Comprobar la veracidad de las hipótesis mediante una encuesta con un grupo cerrado de jugadores.

- Analizar los datos para descartar, añadir o redefinir los valores estudiados de la hipótesis.
- Si se da tal caso, redefinir la hipótesis para que aplique las nuevas variables y permita así al motor categorizar las historias.
- Comprobar el correcto funcionamiento del motor mediante una encuesta con otro grupo distinto de jugadores.
- Analizar los resultados obtenidos.
- Extraer las conclusiones finales en base a dichos resultados.

1.4 Metodología y plan de trabajo

Al inicio del proyecto, el grupo se plantea reuniones semanalmente para estudiar el estado del arte sobre Narrativa Computacional, Creatividad Computacional y Narrativa en general, con el objetivo de definir los parámetros a estudiar y las tecnologías que se usarán para llevar a cabo el proyecto. La información recabada se analiza en el **Capítulo 2: Estudio del trabajo previo**.

Una vez definida la hipótesis de partida y los objetivos a lograr, se comienza el desarrollo del prototipo de juego en Unity que servirá para analizar la veracidad de la hipótesis. Paralelamente, se implementa una primera versión del motor en C# que se limita a almacenar los datos de las historias y ofrecer una serie de funcionalidades al prototipo de juego para que este pueda cargar las historias. Se decide aumentar la frecuencia de reuniones semanales a una cada dos días, al resultar más eficaz para el desarrollo que los miembros del grupo debatan sobre las implementaciones más complejas y trabajen siguiendo el modelo de "Pair programming".

Tras la implementación del prototipo, se redacta una encuesta en Google Forms y se publica el juego en itch.io¹ para que un grupo limitado de jugadores lo prueben y se pueda comprobar si la hipótesis planteada es correcta. Los datos obtenidos se analizarán para aceptar o refutar la hipótesis y replantearla si fuese necesario.

Con la hipótesis revisada, se define la implementación del proceso de identificación que realizará el motor a partir de los parámetros de cada historia. Con el objetivo de realizar

¹ <https://itch.io/>

una segunda encuesta a otro grupo distinto de jugadores, se desarrolla el motor y se adapta el prototipo de juego a los cambios realizados. De nuevo, se publica la versión completa en itch.io y se redacta una encuesta para los jugadores que se analizará para extraer las conclusiones del proyecto.

1.4.1 Fases del proyecto

A continuación se incluye la figura 1 que refleja las diferentes fases del proyecto junto a la enumeración de las diferentes tareas desarrolladas en cada una de estas fases.

Fase del proyecto	Descripción del trabajo
Octubre de 2020 Planteamiento del proyecto	Búsqueda de recursos: documentación para la investigación, recursos y herramientas. Planificación: organización del trabajo de cada fase.
Noviembre de 2020 Diseño de los sistemas base del juego	Diseño de sistemas: jugabilidad y estructura de clases. Diseño narrativo: historia principal y mapa del juego. Memoria: Registro de diseños.
Diciembre de 2020 Desarrollo del juego	Implementación del juego: sistema de clases, gestión de historias y tileset. Memoria: Registro de los sistemas y diseño estético.
Marzo de 2021 Planteamiento del prototipo del Motor	Planteamiento de hipótesis: variables de estudio iniciales y diseño de historias que la cumplan. Diseño del motor: estructura de clases y archivos de datos con el desarrollo de las diferentes historias. Memoria: Registro de los diseños e hipótesis.
Abril de 2021 Implementación del prototipo del motor	Implementación del motor: gestores y clases, sistema de carga de ficheros y peticiones con Unity. Creación de los ficheros con la información del juego. Memoria: Registro del funcionamiento del motor.
Mayo 2021 Experimentación y estudio de hipótesis	Lanzamiento del proyecto y formulario de resultados. Análisis de los resultados y reajuste de la hipótesis. Memoria: Registro de resultados y conclusiones finales.
Junio 2021 Diseño del motor de procesado	Diseño del sistema de clasificación de historias y sistemas que analizan las variables de la hipótesis. Memoria: Registro del planteamiento de los sistemas.
Julio de 2021 Desarrollo del motor	Implementación de sistemas: análisis de las variables de las hipótesis y clasificación de historias. Memoria: Registro del sistema de clasificación de historias.
Agosto de 2021 Experimentación y Testing	Lanzamiento final con formulario de validez de la hipótesis. Análisis de los datos para plantear una conclusión. Memoria: Registro de resultados y conclusión final.

Figura 1: Planificación por fases del proyecto

1.5 Estructura del resto del documento

En el **Capítulo 2: Estudio del trabajo previo** se habla del estudio realizado previamente para poder abordar este proyecto. Se menciona toda la bibliografía encontrada para la adquisición de las herramientas necesarias para poder realizar la investigación y las herramientas utilizadas para ello.

Más adelante se procede a explicar en los apartados “Diseño del prototipo de juego y contexto narrativo”, “Diseño Computacional”, “Implementación”, “Pruebas del prototipo”, “Aplicación de cambios tras las pruebas” y “Pruebas finales” todo el trabajo realizado. La primera fase explicada en el **Capítulo 3: Diseño del prototipo de juego y contexto narrativo** y **Capítulo 4: Diseño computacional del motor de procesado de historias** será el desarrollo de la implementación del sistema en Unity que utilizará el motor y prototipo de este. En el desarrollo en Unity, se hablará de cómo se han organizado las clases y de las opciones de las que dispone el jugador junto a una explicación de los diferentes sistemas que ofrece tanto a nivel mecánico como estético.

Al finalizar dicho apartado, se encuentra el **Capítulo 5: Implementación**, donde existe un desarrollo de la estructura interna del motor junto a los sistemas y clases encargados de almacenar la información procedente de los archivos descritos en los anteriores capítulos. También se habla del funcionamiento de este dependiendo de las diferentes fases de ejecución en las que se encuentre el juego.

Una vez definido el funcionamiento del motor, se desarrollan en primer lugar las pruebas realizadas para analizar el estado de la hipótesis, analizando las variables establecidas en ella y los avances que se van obteniendo en el **Capítulo 6: Pruebas del prototipo**. Más adelante se desarrolla una explicación de cómo se comprueba a nivel computacional la influencia de cada variable de la hipótesis reajustada en el **Capítulo 7: Aplicación de cambios tras las pruebas** y, por último, se desarrolla el nuevo experimento para comprobar el funcionamiento del motor en el **Capítulo 8: Pruebas finales**.

A estos apartados, tanto las pruebas de funcionamiento del prototipo como de la versión final del motor, se le adjuntará un apartado de resultados obtenidos con nuestros descubrimientos y los datos obtenidos de encuestas realizadas a los jugadores que prueben el juego.

Tras estas pruebas, se analizan los resultados de todas las pruebas realizadas para analizar los diferentes datos que estos ofrecen y se analizan en conjunto con el desarrollo de todo el proyecto en el **Capítulo 9: Discusión** y una conclusión basada en este apartado y el trabajo futuro aplicable a este proyecto, que se desarrollan en el **Capítulo 10: Conclusiones y trabajo futuro**.

Por último, se encuentran desarrolladas en el **Capítulo 11: Contribución individual** las contribuciones individuales de cada integrante del equipo durante el progreso del proyecto.

Como aportación adicional, las versiones en inglés del resumen, la introducción y las conclusiones se encuentran en **Anexo 1: Abstract**, **Anexo 2: Introduction** y **Anexo 3: Conclusions and future work**.

Capítulo 2: Estudio del trabajo previo

2.1 Sistemas de procesamiento automático de narrativa

Para empezar con el trabajo se han recopilado una serie de artículos, libros y conferencias sobre Narrativa, Narrativa Computacional y Creatividad Computacional. Los sistemas inteligentes desarrollados han permitido al grupo entender el estado actual de las tecnologías de Inteligencia Artificial aplicadas a narrativa, las técnicas empleadas en los distintos proyectos con sus ventajas y desventajas, y la estructura y componentes de las narrativas, tanto aplicadas a videojuegos como en general. Estos datos han aportado el conocimiento necesario para definir las variables cuantificables más relevantes y proponer la hipótesis de partida.

Para situar el contexto de la investigación, se ha redactado un apartado sobre el desarrollo de la narrativa en los videojuegos a lo largo de la historia, comentando diversos videojuegos y sus diferentes aproximaciones narrativas. El objetivo es ver el progreso de la narrativa en los videojuegos y cómo la categorización en historias principales y secundarias ha ido evolucionando.

2.1.1 Sistemas de gestión de la narrativa

“La investigación en inteligencia artificial y los videojuegos son una combinación natural, y el mundo académico es un lugar fértil para combinar la producción de juegos y la investigación académica” (Gold, 2005). Las inteligencias artificiales (IA) de los juegos y las usadas en los ámbitos académicos tienen una relación simbiótica casi desde sus inicios. Es cierto que la finalidad de una IA en el mundo de la investigación es diferente a la de los videojuegos, sin embargo, debido al alto número de posibles jugadores es común que los videojuegos se valgan de las técnicas del ámbito académico y este, a su vez, beba de los descubrimientos y avances que los juegos hayan realizado.

Un destacable ejemplo sobre cómo un videojuego ayudó a avanzar la investigación de IA es el juego *Black & White* (Lionhead Studios, 2001), un juego de simulación y estrategia cuya innovadora IA logró un Récord Guinness (*Artificial Intelligence and Interactive Entertainment Conference*, 2007) en 2001. En el juego existen unos seres llamados “criaturas”, los cuales basan su aprendizaje y toma de decisiones en el modelo

creencia-deseo-intención de Michael Bratman (Bratman, 1987). Una criatura forma una intención combinando deseos, opiniones y creencias; los deseos son las metas que la criatura quiere cumplir, las opiniones las diferentes formas de satisfacer un deseo y las creencias la información del mundo. Para cada deseo, la criatura elige la creencia con la mejor opinión, generando así una intención o meta a cumplir.

El objetivo de Peter Molyneux, diseñador principal de Black & White, era crear una IA que fuese capaz de pasar el test de Turing. Para ello, su equipo colaboró con distintas instituciones académicas como el Instituto de Tecnología de Massachusetts (MIT) (Gray, 2001).

El test de Turing es una prueba de inteligencia y personalidad en el que se interroga a un humano y a una computadora a través de texto, no pudiendo así conocer la naturaleza de cada uno de los interrogados (humano o máquina). El objetivo de dicho experimento es conocer si el interrogador es capaz de discernir quién es la máquina. Si la persona que realiza las preguntas es incapaz de diferenciar a la máquina del humano, la prueba se da por exitosa (*Alan Turing Scrapbook - Turing Test*, 1950). Pese a la gran fama del test ideado por Turing hoy en día ninguna IA ha podido superar la prueba.

La inteligencia artificial es un amplio campo de estudio con multitud de ramas de investigación, entre ellas la Creatividad Computacional. En el ámbito de este proyecto, se profundiza en el área de la Narrativa Computacional, que a su vez es parte del campo de la Creatividad Computacional.

Según Tony Veale, Amílcar Cardoso y Rafael Pérez y Pérez: *“La Creatividad Computacional (CC) es una rama emergente de la IA que estudia y explota el potencial de las computadoras de ser más que herramientas con multitud de funcionalidades, y de actuar como creadores autónomos o co-creadores. En un sistema CC, el ímpetu creativo debería provenir de la máquina, no el humano, aunque en sistemas CC híbridos un ímpetu conjunto puede provenir de ambos”* (Veale et al., 2019).

Por su parte, la Narrativa Computacional hace uso de la inteligencia narrativa, que podríamos definir como: *“... es la habilidad para construir, narrar, entender y responder de forma afectiva a historias”* (Riedl, 2016). De esta forma, ambos conceptos quedan relacionados en el término de inteligencia narrativa computacional: *“La investigación en*

inteligencia narrativa computacional busca inculcar inteligencia narrativa en computadoras. Al hacerlo, el objetivo de desarrollar inteligencia narrativa computacional es hacer a las computadoras mejores comunicadores, educadores, artistas y más capaces de relacionarse con nosotros al entender genuinamente nuestras necesidades. La inteligencia narrativa computacional trata tanto de la interacción humano-computadora como de resolver problemas complejos de inteligencia artificial” (Riedl, 2016).

Para la creación de contenido de forma autónoma, en el ámbito de los videojuegos, definimos la generación procedural de contenido como: “... la creación automática de contenido usando algoritmos” (Togelius et al., 2011). Consecuentemente, la generación procedural de narrativa es la creación automática de historias usando algoritmos, teniendo en cuenta que la generación se produce por completo durante la ejecución del sistema aunque existan precondiciones establecidas por los diseñadores o programadores. Estos sistemas de generación reciben el nombre de Sistemas de Generación de la Narrativa.

A continuación, se estudiarán las principales aproximaciones a la generación procedural de narrativa haciendo referencia a la conferencia en la Game Developers Conference impartida en 2017 titulada Procedural Narrative Generation por parte de Rogelio Cardona-Rivera y Chris Martens. (Cardona-Rivera & Martens, 2017)

“El principal reto a la hora de diseñar estos sistemas es mantener el equilibrio entre la espontaneidad y la estructura” (Cabrero Martín et al., 2021). Otorgar mucha libertad al jugador supone unos mayores valores de potencial creatividad por parte del sistema, dándose situaciones que se pueden salir de lo planeado por los diseñadores pero favoreciendo la rejugabilidad y el contenido. Por otro lado, favorecer la estructura narrativa que diseña el autor acota las posibilidades del sistema, ya que este debe estudiar si las decisiones del jugador se salen de los límites establecidos para reconducir la trama a los objetivos marcados.

Por tanto, diferenciamos las dos aproximaciones más comunes a la generación procedural de narrativa: centrarse en la simulación o centrarse en la trama. Es necesario definir formalmente los elementos que participan en la generación, tanto los personajes como las posibles circunstancias que se pueden afectar a estos, para que el sistema pueda operar con ellos.

El primer modelo estudiado es el centrado en la simulación, que además sirve de base para el modelo centrado en la trama. El modelo centrado en la simulación emplea una serie de reglas de interacción y un estado inicial, diseñados por el autor o diseñador, que son aplicados por un agente inteligente sobre un conjunto de personajes. La simulación puede ser totalmente autónoma o incluir la interacción del jugador. En la figura 2 se puede ver una representación gráfica de este modelo.

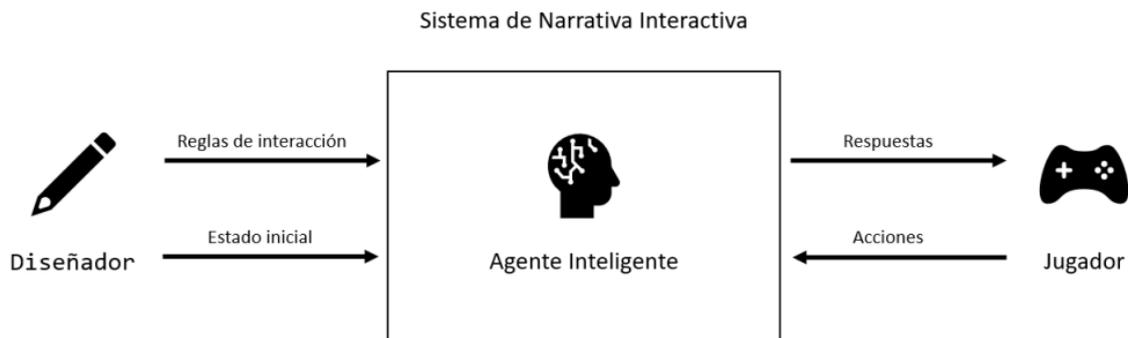


Figura 2: Conceptualización del acercamiento centrado en simulación para el diseño de un sistema de narrativa interactiva (Cabrero Martín et al., 2021)

El estado inicial es la información del mundo en el que se sitúa la historia, tanto información de los agentes inteligentes como del entorno y los objetos. Es el punto de partida de la historia, la precondition siempre cierta que precede a la simulación y, por lo tanto, debe ser diseñada y estudiada teniendo en cuenta las limitaciones del sistema y las posibilidades de interacción.

Las reglas de interacción describen las acciones que pueden ejecutar los agentes inteligentes o el jugador para alterar el estado de la simulación. Las reglas pueden ser tanto específicas a ciertas condiciones como generales, siendo estas últimas las más recomendables al aportar un mayor grado de aleatoriedad y ramificación. De nuevo, es el diseñador quien debe decidir las reglas y cómo afectan a los estados de la historia.

A partir de los personajes y sus reglas de interacción, se crea un espacio de estados que contiene los diferentes estados por los que puede pasar la simulación. Formalmente, se puede visualizar como un grafo en el que los vértices son estados y las aristas son las condiciones que deben darse, mediante la ejecución de reglas de interacción, para pasar del

estado actual al siguiente. Como resultado, una simulación generará un camino entre estados que conforman una historia.

Como se ha comentado anteriormente, este modelo es susceptible a las interacciones del jugador, pues el diseñador puede verse incapaz de anteponerse a todas las decisiones que los jugadores puedan tomar y no estén contempladas. Como resultado, se pueden generar estados imprevistos a los que se darían en una simulación totalmente autónoma. Para intentar subsanar este problema, se puede limitar la capacidad de interacción del jugador, por ejemplo permitiéndole controlar solo a un personaje de todos los presentes en el mundo, o controlando el mundo en lugar de a los personajes.

La alternativa, que permite mayor control sobre el progreso de la narrativa según se ha diseñado de antemano, son los sistemas centrados en la trama. Partiendo de la base comentada anteriormente de los sistemas centrados en la simulación, ampliamos la arquitectura con un componente extra denominado la agenda narrativa. En la figura 3 se puede ver una representación gráfica de este modelo.

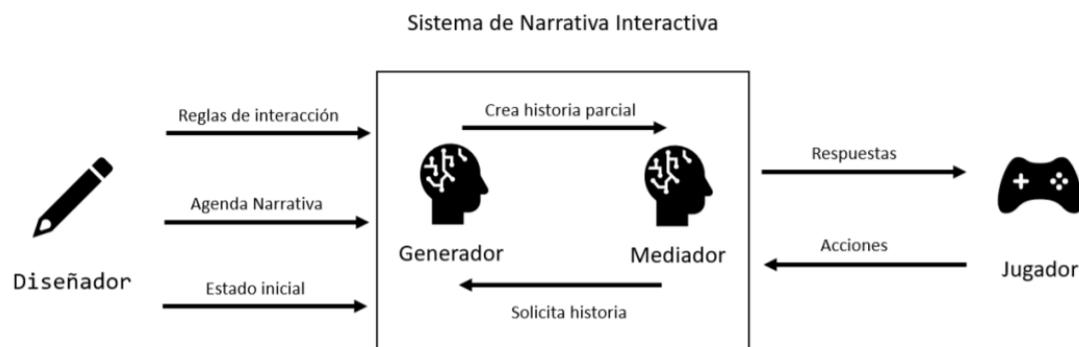


Figura 3: Conceptualización del acercamiento centrado en la trama para el diseño de un sistema de narrativa interactiva (Cabrero Martín et al., 2021)

La agenda narrativa se compone de los objetivos que el diseñador quiere que se cumplan, por lo que la responsabilidad de su definición recae en él. Para que el sistema cumpla con la agenda narrativa, se añade otro agente inteligente en la arquitectura denominado mediador, el cual debe monitorizar las acciones que lleva a cabo el jugador para redirigir a los objetivos establecidos.

En este caso, los caminos que se producen en el espacio de estados están más acotados, pues la definición de la agenda narrativa obliga a que se cumplan una serie de

condiciones y estados en una historia dada. Estos estados obligatorios restan flexibilidad a la capacidad de generación del sistema, pero facilitan que las interacciones del jugador sean más previsibles y se mantenga una cierta coherencia narrativa. Si el jugador intenta alejarse de los estados marcados en la agenda, el sistema debe estar preparado para o bien impedir que el jugador tome decisiones en esa dirección, o bien se generen historias adicionales que reconduzca la historia a los estados objetivo.

Para complementar los conocimientos adquiridos sobre generación de narrativa, se realiza un repaso de algunos sistemas de gestión de la narrativa que han realizado contribuciones relevantes a la investigación de la narrativa computacional (Gervas, 2009).

Novel Writer

Novel Writer (Klein et al., 1973) es el primer sistema de gestión de la narrativa del que existen registros. Fue creado específicamente para cumplir con un único género literario, las historias de misterio y asesinato en el contexto de fiestas de fin de semana, con una longitud máxima de 2100 palabras y un tiempo de generación inferior a los 19 segundos (en su momento de publicación).

Empleando un modelo de microsimulaciones, el comportamiento y las acciones de los personajes se basan en reglas probabilísticas que alteran de forma progresiva el estado de la simulación global. Como la historia debe ajustarse al género de misterio y asesinatos, el flujo de la trama siempre sigue una secuencia que cumple con lo esperado y las reglas de interacción entre personajes es muy restrictiva, por lo que no existe una gran diferencia entre las distintas historias generadas. El sistema requiere de varios datos de entrada para conformar el estado inicial, como una descripción del mundo en el que tiene lugar la historia o los rasgos del asesino y la víctima.

Toda la información se representa en forma de una estructura semántica que toma forma de red, es decir, un grafo en el que los nodos son objetos semánticos y las aristas son las relaciones que unen dichos objetos, de forma similar al espacio de estados que se ha comentado. La aplicación de reglas de interacción produce cambios en los estados y hace progresar la simulación, traduciendo los sucesos a lenguaje natural para informar del estado de la historia al lector.

Talespin

Talespin (Meehan, 1977) es un sistema que genera historias sobre animales de bosque corrientes, con la capacidad de realizar tanto encadenamiento hacia adelante (forward chaining) como encadenamiento hacia atrás (backward chaining).

El encadenamiento hacia adelante y el encadenamiento hacia atrás son los métodos principales de razonamiento cuando se utiliza una máquina de inferencia, un componente de sistemas de inteligencia artificial que aplica reglas lógicas a la base de conocimientos para deducir nueva información. El encadenamiento hacia adelante se puede describir como la aplicación repetida de *modus ponens*, una forma de razonamiento deductivo que se puede resumir como “si P implica Q; y si P es verdad; entonces Q también es verdad”. El encadenamiento hacia atrás, por su parte, hace uso de la inducción hacia atrás, el proceso de razonar atrás en el tiempo, desde el final de un problema o situación, para determinar una secuencia de acciones óptimas.

El sistema crea objetivos o metas para los personajes, los cuales se traducen en acciones que dichos personajes deben tomar para solucionarlos. Los objetivos se pueden dividir en subobjetivos, y se puede dar más de un elemento que cumpla el objetivo en una misma historia, por lo que el sistema crea una lista de subobjetivos para cada elemento. También se tiene en cuenta la percepción de los personajes, es decir, de lo que son conscientes en su entorno y de lo que no, y se profundiza en las relaciones entre personajes, permitiendo que sean más complejas y que afecten a las acciones que los personajes toman entre ellos.

Author

Author (Dehn, 1981) fue diseñado para simular la mente del autor durante el proceso de creación de una historia, modelando los factores involucrados como los personajes, sus relaciones, su situación en el mundo o los momentos relevantes que se dan en este.

El razonamiento de Dehn es que los autores tienen unas ideas iniciales de lo que quieren que ocurra en una historia, o una serie de eventos que formarán la base de la trama, y construyen el resto de la historia a partir de esto. Por lo tanto, cada autor define sus propios objetivos a lograr en la redacción y estos objetivos a su vez se dividen en subobjetivos. Estos objetivos pueden ser más abstractos, como asegurar que la narrativa es

consistente y plausible, que los personajes sean creíbles y que se mantenga el nivel adecuado de drama.

Universe

Universe (Lebowitz, 1983) es capaz de generar una sucesión de scripts para telenovelas, en los que una multitud de personajes forman parte de diversas historias simultáneas sin un final establecido.

El foco principal del sistema es la generación y evolución de los personajes, aportando complejas estructuras de datos para modelarlos, aunque la mayor parte de los datos deben ser introducidos manualmente por el autor. Esto se ve apoyado en la creencia de Lebowitz de que es más interesante crear previamente el mundo donde se desarrolla la historia, en contraposición con el acercamiento de Dehn de crear primero la trama y a partir de ahí el mundo.

La generación que lleva a cabo *Universe* se centra en identificar metas en la trama y hacer que los personajes tomen las acciones necesarias para cumplir dichas metas, aunque las acciones no sean las que hubiesen tomado de forma independiente. El sistema mantiene un grafo de precedencia con la información de los objetivos, los fragmentos de trama y su relación entre ellos, de forma similar a la planificación descomposicional, para planificar los siguientes pasos de la trama.

Es interesante estudiar cómo *Universe* introduce un mecanismo que le permite ampliar su biblioteca de fragmentos de trama de forma automatizada, creando nuevos fragmentos. Este proceso se lleva a cabo generalizando fragmentos existentes e instanciando la estructura resultante para construir otros nuevos, comprobando que no se produce pérdida de información al compararlo con el fragmento inicial. Gracias a este proceso, se han identificado objetivos fundamentales tales como “mantener la tensión romántica de los personajes” o “hacer que la historia avance”.

Minstrel

Minstrel (Turner, 1993) genera historias sobre el Rey Arturo y sus caballeros de la mesa redonda, usando una moraleja como “semilla” o “seed” para construir la historia. Necesitaba de historias preexistentes para generar nuevas historias, siendo la versión

original capaz de crear unas 10 historias de una longitud de entre media y una página, y un mayor número de historias más cortas.

El sistema usa un modelo denominado unidades de construcción, compuestos por objetivos y planes que los cumplen, los cuales operan al nivel de objetivos de autor y objetivos de personajes. La generación se separa en los niveles de planificación y resolución de problemas, siendo el primero el que opera con los objetivos de autor, dividiéndolos en subobjetivos o pasándolos al nivel de resolución de problemas. Este nivel estudia los objetivos y añade los elementos necesarios para poder cumplirlos, comprobando que se cumple con la consistencia de la trama.

Agent Stories

Agent Stories (K. Brooks, 1995) (K. M. Brooks, 1997) es un sistema orientado a la secuenciación de historias y la orquestación, capaz de generar historias no lineales con múltiples puntos de vista. El objetivo es que este sistema sirva de complemento al autor, asistiendo en la creación de tramas no lineales.

El acercamiento que hace el sistema hace uso de tres componentes de la narrativa computacional: la estructura de la historia; la colección y organización de las partes de la historia con alguna representación de su significado; y una estrategia de navegación por la colección de partes de la historia que tenga estilo y propósito. A su vez, se divide el proceso en tres partes: el entorno estructural, en el que se define la estructura de la historia; el entorno del escritor, en el que se captura y representa el conocimientos de las partes de la historia; el entorno del editor, en el cual agentes software secuencian las diferentes partes de la historia de acuerdo a las preferencias estilísticas de cada agente.

Los agentes software son los encargados de la generación de las historias, siendo interesante la posibilidad de definir estilos a cada agente de forma que la generación está acotada a dichos estilos definidos. Además, los agentes aprenden de las generaciones realizadas gracias al feedback proporcionado por el diseñador, de forma que las siguientes generaciones se ajustan más a la visión y las expectativas del diseñador.

Mexica

Mexica (Pérez y Pérez, 1999) es un sistema diseñado para el proceso creativo de escribir siguiendo el ciclo de *engagement* y *reflection*, y para ello genera historias breves sobre los nativos mexicanos.

Durante la fase de *engagement* se añade una acción a la trama, si se cumple que algún contexto del mundo de la historia, o *story-world context* (SCW), enlaza la trama con esa acción. En la fase de *reflection* se repasa la trama hasta el momento, comprobando la coherencia, la novedad y el interés que genera.

Los SCW son un esquema especial que usa *Mexica* para almacenar la información de historias anteriores. Representan instancias de contextos, como enlaces emocionales entre personajes, en los cuales se ha dado una acción en historias anteriores, actuando como reglas durante la fase de *engagement*.

The Virtual Storyteller

The Virtual Storyteller (Theune et al., 2003) se concibe como una propuesta intermedia entre la generación centrada en la simulación y la centrada en la trama, otorgando a los personajes de la historia su propio conocimiento sobre el mundo y sus propias reglas de interacción pero añadiendo al modelo un agente director que supervisa el progreso de la trama.

El agente director tiene los conocimientos necesarios para dirigir la trama, como que esta debe tener un planteamiento, un nudo y un desenlace (feliz), y ejerce control sobre el resto de personajes de tres formas posibles: mediante el entorno, introduciendo nuevos personajes u objetos; mediante motivaciones, dando objetivos a los personajes de la historia; y mediante restricciones, impidiendo que personajes lleven a cabo ciertas acciones. En cambio, no puede forzar a los personajes a realizar ciertas acciones.

A diferencia de los sistemas anteriores, *The Virtual Storyteller* incluye un agente narrador que traduce los estados del sistema a lenguaje natural, con principal énfasis en el uso de pronombres para evitar la repetición constante de nombres. Como resultado, el texto parece más natural y atractivo de leer.

Fabulist

Fabulist (Riedl & Young, 2010) es un sistema que genera y presenta historias de forma automatizada en el formato de fábulas. Destaca por su uso del algoritmo de planificación *intent-driven partial order causal link* (IPOCL), que estudia la causalidad, intencionalidad y motivación de los personajes para producir secuencias narrativas que son causalmente coherentes, otorgando a los personajes cierta credibilidad.

La arquitectura divide la generación en tres niveles: la generación de la fábula, donde se aplica el anteriormente comentado algoritmo IPOCL; la generación del discurso narrativo; y la representación visual de la historia. Además, para asegurar que se cumple con el estado final que se puede introducir como entrada al sistema, se realiza al inicio de la ejecución un plan narrativo teniendo en cuenta a todos los personajes y los objetivos de la historia, aunque también es capaz de realizar los cambios necesarios en los datos de entrada si con los originales no es posible cumplir los objetivos.

Narrador en nn

El sistema *nn* (Montfort, 2007) de ficción interactiva permite a un usuario interactuar con el sistema mediante descripciones cortas de lo que quiere que haga su personaje. La respuesta dada describe las consecuencias de las acciones del personaje, haciendo uso del lenguaje natural a un nivel más preciso que los sistemas anteriores.

El narrador *nn* es capaz de narrar los hechos en un orden no cronológico, empleando *flashbacks* (analepsis) y *flashforwards* (prolepsis), además de narrar en orden directo o inverso o incluso entrelazando sucesos de diferentes momentos. También está diseñado para usar los tiempos verbales adecuados según el momento del suceso y emplea los denominados *focalizer worlds* (mundos focalizadores) para focalizar o destacar segmentos considerados relevantes de la narrativa.

2.1.2 Narrativa en los videojuegos

La narrativa ha recibido varias definiciones por parte de narratólogos, que denotan un fuerte consenso sobre la naturaleza del objeto de estudio de su disciplina. Genette la define como “la representación de un evento o una secuencia de eventos” (Genette & Levonas, 1976), mientras que Prince la define como “La representación [...] de uno o más eventos, reales o ficticios, comunicados por uno, dos o más [...] narradores [...] a uno, dos o más

lectores” (Prince, 2003). De ambas definiciones se puede deducir que los eventos son el componente principal de la narrativa, y, siguiendo las palabras de Prince, también se puede añadir la necesidad de un narrador y un lector.

Traspassando estas definiciones al ámbito de los videojuegos, el narrador corresponde al propio juego o al sistema que maneja el estado de la historia, mientras que el lector es el jugador que forma parte de la historia.

Por otro lado, cabe destacar el análisis que realiza Marie-Laure Ryan (Ryan, 2007) sobre estas definiciones:

“Una definición debe respaldar, incluso implicar, declaraciones como estas, pero no es necesario detallarlas:

La narrativa trata sobre la resolución de problemas.

La narrativa trata sobre el conflicto.

La narrativa trata sobre las relaciones interpersonales.

La narrativa trata sobre la experiencia humana

La narrativa trata sobre la temporalidad de la existencia.”

Los elementos de estas declaraciones citadas anteriormente se encuentran en la narrativa de los videojuegos de aventuras tanto actuales como clásicos.

Colossal Cave Adventure (Crowther & Woods, 1976), es un juego basado en texto considerado como el primer videojuego de aventuras. En él, el jugador debe recorrer una cueva para encontrar una serie de tesoros y sobrevivir, consiguiendo la mayor cantidad de puntos posibles. La historia del juego es narrada de forma similar a una partida de Dungeons & Dragons (Gygax & Arneson, 1974), en la que existe un Dungeon Master que actúa como narrador y un jugador que actúa como lector. Asimismo, vemos que se cumplen varios de los puntos comentados por Marie-Laure Ryan, como la resolución de problemas (el jugador debe resolver desafíos para obtener los tesoros) o el conflicto (además de enfrentarse a criaturas, el jugador debe tomar la mejor decisión para conseguir la mayor cantidad de puntos).

No obstante, se puede ver que existe un elemento que diferencia la narrativa clásica de la narrativa en los videojuegos: la interacción con el entorno y la capacidad del jugador para alterarlo. Si bien existen experiencias alternativas de narrativa interactiva como los libros de 'Elige tu propia aventura' (Packard, 1979) o juegos de mesa como Gloomhaven (Cephalofair, 2017), estas están limitadas en posibilidades al tener sistemas de juego más acotados. Un videojuego puede explotar la capacidad de cómputo de un ordenador o consola para que sistemas de inteligencia artificial analicen la narrativa y la situación del jugador, generando historias con una mayor variabilidad.

Por otro lado, también vemos casos de videojuegos en los que no existe una narrativa definida, sino que se le dan al jugador las herramientas y mecánicas para crear su propia historia y el juego reacciona a sus decisiones. Rimworld (Ludeon Studios, 2016), un simulador de colonias espaciales, crea un mundo generado proceduralmente en cada partida, incluidos los personajes y las facciones que lo habitan. Aunque se comunica al jugador que el objetivo de la colonia es encontrar una forma de escapar del planeta, constituyendo como historia principal la decisión que toma el jugador para cumplir con dicho objetivo y los pasos que ha de dar hasta lograrlo, el sistema de juego genera "eventos" para dificultar o acelerar el progreso de la colonia, como asaltos de saqueadores o cápsulas espaciales llenas de suministros. Estos "eventos" crean pequeñas narrativas en las que los colonos pueden tener crisis nerviosas, establecer relaciones (amistosas o no) entre ellos o incluso morir, causando pesar a sus seres queridos o felicidad a sus enemigos. Es el propio sistema de juego el que estudia la situación actual de la colonia para decidir el tipo de evento y la gravedad, e incluso se puede escoger el tipo de "narrador", siendo este el sistema inteligente generador de eventos, de forma que unos favorecen más un progreso lineal, otros otorgan mucho espacio entre eventos para que el jugador pueda reconstruir y progresar, y otros aleatorizan los eventos para dar lugar a situaciones más inesperadas.

Pasando este sistema a los diferentes tipos de historias que se han comentado, tenemos la narrativa principal por la que la colonia debe progresar y crecer para cumplir el objetivo de escapar del planeta y, a su vez, multitud de narrativas que suponen desafíos al jugador, pero que tienen la posibilidad de hacer progresar la colonia hacia dicho objetivo final.

Este videojuego es digno de estudio porque, aunque todos los sucesos que ocurren en la partida se podrían considerar parte de la historia principal, deduciendo entonces que

forman parte de una única historia, hay eventos más relevantes que otros, ya sea porque suponen un mayor riesgo o porque otorgan una mayor ventaja. Al final, recae sobre el propio jugador decidir cuáles son los objetivos intermedios que forman parte de la historia principal, es decir, que más relevancia tienen en la partida y acercan a cumplir el objetivo final del juego, y cuáles son objetivos secundarios, menos relevantes o con menores recompensas.

Otro ejemplo que ahonda más en la ausencia de una narrativa definida es Los Sims (Maxis, 2000), un simulador de la vida real en el que los jugadores pueden crear personajes, o "sims", y guiar sus vidas, desde su nacimiento hasta su muerte, decidiendo sus estudios, trabajos, amistades, etc. Es un juego que no tiene un objetivo principal definido, y toda la responsabilidad de decidir la narrativa recae en el jugador, quien puede decidir que su meta puede ser desde ser lo más rico posible a ser el más famoso de la ciudad pasando por formar una gran familia.

Sin embargo, el juego sugiere ciertos objetivos en cuanto a los deseos propios de cada "sim", que al cumplirlos hacen que este sea más feliz. Su cumplimiento no es obligatorio, es decir, depende de si el jugador desea que dicho "sim" consiga felicidad, pues una partida en la que todos los "sims" son infelices es completamente plausible. De esta forma, la definición de historias principales y secundarias se vuelve totalmente subjetiva y ajena al juego.

Estos juegos comentados pertenecen al género "sandbox", o "caja de arena", referenciando a las construidas en parques infantiles para que los niños jugasen con su imaginación al crear cualquier tipo de objeto hecho de arena. "Los videojuegos jugados en modo sandbox, o juegos específicamente diseñados como sandbox, permiten al jugador acceder al mundo por completo (la caja), y permiten que pueda cambiarlo a voluntad (la arena)" (Lange, 2018).

La libertad otorgada al jugador en este tipo de juegos, incluida la escasez de narrativa que guíe sus decisiones, resulta en una mayor dificultad para estudiar los componentes narrativos. "Los elementos de sandbox pueden considerarse erróneamente como reemplazos adecuados al contenido narrativo; efectivamente, multitud de juegos han perdido su potencial al pensar que la libertad de juego compensaría una falta de narrativa"

(Breslin, 2009). Por estos motivos, este es un género que puede beneficiarse de una potente generación de historias que incremente la rejugabilidad y el contenido.

Otros géneros ofrecen un acercamiento más estándar de la narrativa. *Fallout* (Interplay, 1997) es un juego de rol clásico ambientado en un mundo postapocalíptico en el que al jugador se le presenta la trama principal desde el momento en el que se inicia la partida: encontrar un chip de agua para que su refugio pueda seguir teniendo agua potable.

Aunque se le otorga al jugador la libertad de explorar el mapa y realizar las historias secundarias al ritmo y orden que desee, la diferencia más importante que encontramos entre la principal y las secundarias consiste en que una vez pasados una cantidad de días en el juego, el refugio es destruido y la partida termina. Esta forma de concluir la historia es considerada como experimentar un “final malo” ya que implica que muchas incógnitas de la historia principal quedarán sin desvelar. Asimismo, como otra forma de exhortar a los jugadores a que sigan la historia principal, se resalta la importancia de este desde el principio al poner en una situación crítica a los allegados del protagonista.

Unidas ambas situaciones, tanto el peligro para los seres queridos del protagonista como la posibilidad de la destrucción del refugio, las misiones secundarias quedan relegadas a un segundo plano, aunque traten también de situaciones de vida o muerte o con objetos importantes, principalmente por la desvinculación con el protagonista y la longitud de la narrativa. La historia principal requiere viajar por buena parte del mapa y cumplir multitud de objetivos, mientras que las secundarias se centran en espacios localizados, como ciudades, o requieren de viajes cortos. Estas secundarias amplían el trasfondo del juego, pero no afectan a la resolución de la misión principal.

Como contraposición, *Mass Effect* (Bioware, 2007) es una trilogía de juegos de rol y acción de ciencia ficción en los que las secundarias, aún siendo menos relevantes que la trama principal, afectan al desarrollo y conclusión de esta. En la primera entrega, se le da la posibilidad al jugador de reclutar a dos personajes para el equipo, estos son completamente opcionales. Sin embargo, la inclusión de ambos en el equipo influye y ramifica la historia a lo largo de las 2 entregas siguientes, permitiendo diferentes situaciones y decisiones por parte del jugador.

En la tercera entrega sucede una guerra galáctica y el jugador debe recabar apoyos de distintas facciones para poder ganarla. Estos aliados pueden conseguirse tanto en la historia principal como en las secundarias. Para conseguir el mínimo para salir victorioso en la contienda basta con realizar las misiones principales, sin embargo, esto da lugar al considerado “final malo”. No obstante, este tipo de desenlace es diferente al “final malo” de Fallout, ya que no supone que el jugador pierda la partida sino que se dan las peores circunstancias y el final es menos armonioso.

Esta forma de incentivar a los jugadores a realizar las secundarias les añade un valor extra a este tipo de historias. Más allá de los beneficios que un jugador suele obtener de una historia secundaria, como dinero, mejoras en el personaje o ser parte de una narrativa adicional, el jugador tiene más posibilidades de influenciar en el devenir de los acontecimientos principales y acercar a un final más ideal para el jugador.

Este tipo de narrativa secundarias es digna de estudio, pues hace al jugador percibir estas tramas como más relevantes o como parte de la propia historia principal. Considerando esta posibilidad de que puedan ser identificadas como parte de la historia principal, la mayor diferencia entre otras principales sería la obligatoriedad de completitud, pues estas historias no son necesarias para completar la trama principal.

Debido a la disparidad de los distintos géneros y juegos, y a los diferentes tipos de narrativa que utilizan, la definición de factores comunes que se puedan estudiar para la identificación de historias resulta compleja, lo que refuerza el interés por la realización de este proyecto de investigación y el estudio de los datos que se obtengan tras las pruebas con jugadores.

2.2 Tecnologías usadas en el trabajo

Para el desarrollo del proyecto se han empleado una serie de herramientas, librerías externas y software de terceros, tanto para la propia creación del proyecto como para la organización del grupo y del trabajo.

2.2.1 Herramientas de diseño de historias

Con el objetivo de buscar más ejemplos de modelos y estructuras narrativas, se hizo una investigación de diferentes herramientas e implementaciones de narrativa en el ámbito computacional. Puesto que el planteamiento que se buscaba estaba orientado a la estructura que se utiliza en la escritura de novelas y guiones de cine, se buscaron herramientas que aplicaran estas estructuras para ver cómo se organizaban los datos.

Tras este proceso de búsqueda se dio con las siguientes herramientas para usarlas como ejemplo de estructuración narrativa computacional.

*Bibisco*² y *Scrivener*³

En el momento de crear una historia, se han de tener en cuenta muchos factores que afectan al funcionamiento de la misma. Esto se puede conseguir a través de seguir ciertos procesos de creación que ayudan a la persona que se dedica a crear historias a tener todo mucho más claro para evitar inconsistencias dentro de la trama y poder controlar a los personajes que van a intervenir en la misma de manera más eficiente. En este proceso, como mencionan reiteradamente en el podcast *Writing Excuses* (Sanderson et al., s. f.), es necesario plantear una estructura para el proyecto y qué cosas van a suceder, además del orden de las mismas.

Este proceso, normalmente, pasa por crear lo que se conoce como Escaleta, una lista con todas las escenas y capítulos que va a tener una historia para que se pueda organizar todo de forma eficiente. Aunque muchos escritores lo hacen a mano, existen muchos softwares que facilitan todo este proceso, proporcionando al creador una forma eficaz de planificar sus proyectos.

Una de esas herramientas es *Bibisco*, un procesador de texto libre desarrollado en Node.js que permite al escritor no sólo crear historias, sino que también le permite crear dicha estructura para poder diseñarlas y analizarlas en profundidad.

² <https://bibisco.com/>

³ <https://www.literatureandlatte.com/scrivener/overview>

Además de esto, también ofrece un sistema para crear personajes y darles las diferentes características que van a necesitar, proporcionando al autor de una forma rápida y sencilla para acceder a toda la información de un personaje y poder verla de un solo vistazo.

Otro software similar al ya mencionado anteriormente y utilizado con el mismo propósito, es el procesador de textos *Scrivener*, utilizado para la creación y maquetación de historias. Ofrece las mismas funcionalidades que *Bibisco*, organizando las historias en capítulos y escenas.

Por tanto, de ambos programas se ha podido obtener una forma de organizar las diferentes narrativas que puedan ser claras de cara al autor y computacionalmente, ya que al organizar las historias en partes pequeñas y contenidas en otras más grandes flexibiliza mucho la forma de ampliar las historias y de analizarlas.

2.2.2 Herramientas de planificación y control

Para la organización del trabajo por parte de los miembros del grupo se emplean varias herramientas de planificación que permiten definir tareas, asignarlas a miembros y marcar una fecha límite, de forma que todos saben qué está haciendo cada uno, qué pueden hacer tras acabar su tarea actual y no se duplica trabajo. También se incluye la herramienta que se ha usado para el control de versiones del código del proyecto.

*Trello*⁴

En las primeras fases del proyecto centradas en el diseño y desarrollo del videojuego en *Unity* y el prototipo del motor se utilizó el software web *Trello*, utilizado para la administración de las tareas del proyecto.

Este software era conocido para todos los miembros del equipo por la gran cantidad de opciones que ofrece, por lo que acabó siendo la principal herramienta utilizada dedicada a la planificación, control y reparto de tareas. Se crearon las columnas de Backlog, Current Sprint, Doing, Testing y Done para diferenciar los diferentes estados de las tareas, y cada tarea identificada se le asignaba una etiqueta según su tipo, como Programación, Diseño, Documentación, etc. Cuando un miembro terminaba su tarea, se asignaba la siguiente y seguía trabajando.

⁴ <https://trello.com/>

Hojas de cálculo de Google⁵

Para la organización más global del proyecto, como los diferentes objetivos a cumplir y los tiempos estimados, se decidió usar varias *Hojas de cálculo de Google* que permitían una mayor personalización y adaptación a las necesidades.

GitHub⁶

Forja para alojar proyectos utilizando el sistema de control de versiones Git. Se crearon dos repositorios privados, uno para el prototipo de juego en *Unity* y otro para el motor de narrativa. Se ha elegido *Github* por nuestra previa familiaridad con la plataforma, y poder compartir nuestro trabajo sin causar conflictos en el código.

2.2.3 Herramientas de redacción de la memoria y encuestas

A continuación, se exponen las herramientas empleadas para la redacción de la memoria y la creación, envío y análisis de resultados de las encuestas de los experimentos.

Google Docs⁷

Procesador de texto en línea que permite la colaboración en la redacción de documentos. Se ha elegido por la facilidad de la herramienta para modificar el contenido de la memoria y que sea accesible automáticamente para el resto de miembros del equipo, así como el tutor, que ha sido capaz de hacer revisiones detalladas durante todo el proyecto (incluyendo hacer sugerencias o correcciones), gracias a las funcionalidades de la herramienta hechas específicamente para ello.

Google Forms⁸

Software de administración de encuestas en línea que aporta visualización gráfica de los resultados, facilitando el análisis de los datos obtenidos tras las dos pruebas publicadas. Se ha escogido por la fácil accesibilidad de los formularios por parte de los jugadores, y por su funcionalidad de mostrar los resultados de cada pregunta en forma de gráficos y tablas automáticamente.

⁵ <https://www.google.es/intl/es/sheets/about/>

⁶ <https://github.com/>

⁷ <https://www.google.es/intl/es/docs/about/>

⁸ <https://www.google.es/intl/es/forms/about/>

Zotero⁹

Software de gestión de referencias bibliográficas, que incluye una extensión para los principales navegadores con el cual se pueden añadir fácilmente documentos de todo tipo a una biblioteca de referencias. Además, esta biblioteca se puede compartir con el resto de miembros del equipo, agilizando la recopilación de referencias. Se escogió por la facilidad de uso y la comodidad que otorga en la gestión de las referencias.

2.2.4 Herramientas de desarrollo

Para el desarrollo del software del proyecto, el equipo ha optado por las siguientes herramientas de desarrollo.

Unity¹⁰

Motor de videojuegos utilizado para implementar el prototipo de videojuego que se conecta con el motor de narrativa. Nos permite desarrollar ágilmente este prototipo y generar fácilmente un ejecutable para publicarlo y que los jugadores puedan probar el juego sin necesidad de grandes habilidades técnicas. Se escogió por su facilidad y rapidez para crear juegos sencillos, además de experiencia previa de varios miembros con el motor.

Visual Studio 2019 Community¹¹

Entorno de desarrollo integrado desarrollado por Microsoft usado para la implementación de los sistemas en C Sharp (C#) del proyecto, tanto el motor de narrativas como los scripts de Unity, junto a las herramientas .NET para poder compilar y solucionar errores. Se escogió por experiencia previa de todos los miembros del equipo con el entorno.

Visual Studio Code¹²

Editor de código fuente integrado desarrollado por Microsoft usado para implementar las diferentes clases y métodos que se utilizarán en Unity para el correcto funcionamiento del prototipo de juego. Escogido como alternativa a Visual Studio 2019 cuando se carecía de proyecto, o cuando solo se necesitaban ediciones sencillas, pues Visual Studio 2019 es un programa que ocupa mucha memoria.

⁹ <https://www.zotero.org/>

¹⁰ <https://unity.com/es>

¹¹ <https://visualstudio.microsoft.com/es/>

¹² <https://code.visualstudio.com/>

2.2.5 Librerías externas

Se han usado algunas librerías externas para agilizar el desarrollo de ciertas características y funcionalidades del proyecto, que se describen a continuación.

*QuikGraph*¹³

Librería externa empleada para la creación de grafos y que implementa métodos para realizar recorridos u obtener los caminos más cortos entre vértices. Se decidió añadir al proyecto para agilizar la implementación de grafos y del algoritmo de Floyd-Warshall que usa el motor de narrativa con los lugares de las historias.

*SimpleJSON.NET*¹⁴

Librería externa empleada para la lectura y carga de archivos de tipo JSON. Gracias a esta librería se pueden tratar los datos almacenados en esos archivos, ya que no había compatibilidad con el ya existente en el NET5 y Unity. Por ello, trasladamos el proyecto a NetCore3.1 y utilizamos esta librería, la cual cumplía con la misma funcionalidad que necesitábamos.

¹³ <https://github.com/KeRNeLith/QuikGraph>

¹⁴ <https://github.com/mhallin/SimpleJSON.NET>

Capítulo 3: Diseño del prototipo de juego y contexto narrativo

El objetivo del proyecto es crear un sistema capaz de reconocer la naturaleza de las historias que dan vida a un juego basado en el estudio de las variables computacionales que forman dichas historias. Para poder analizar dichas variables, es necesario en primer lugar diseñar una historia principal junto a una serie de historias secundarias que puedan validar o rechazar la hipótesis de partida planteada.

El prototipo de juego es el entorno creado para poder obtener datos de estudio por parte de los jugadores y para establecer una conexión real entre el motor y un juego. En su diseño, se definen los diferentes sistemas necesarios por un lado para cargar en el sistema las historias y, por otro lado, para la correcta interacción del jugador con el entorno y las historias.

Una vez se diseña la base sobre la que probar el motor y las historias que procesa, se diseñan los diferentes sistemas y estructuras de las que requiere el motor para poder cargar y clasificar las historias diseñadas.

3.1 Diseño de historias para el análisis de la hipótesis

Las historias son las encargadas de ofrecer la información necesaria para poder analizar correctamente la influencia de las variables establecidas en la hipótesis definida en el **Capítulo 1.2 Hipótesis de partida** y que permiten diferenciar una historia principal de una historia secundaria.

Por ello, es necesario diseñar, por un lado, una historia principal que cumpla con los requisitos establecidos en la hipótesis planteada en el y, por otro lado, diseñar las historias secundarias que analizan cada una de las variables de la hipótesis mencionada, siendo estas la longitud de la historia, la cantidad de personajes involucrados en ella y el número de lugares recorridos para completarla.

Las variables establecidas para estudiar la clasificación de las historias quedan reflejadas en cada una de las tramas, por lo que se redacta un guión del juego para poder

analizar la intervención de dichas variables para que cada una de las historias cumpla con los requisitos que necesita para estudiar la involucración de las variables mencionadas en la hipótesis de partida establecida en el .

Por ello, se define en primer lugar una historia general de la que poder desarrollar las diferentes historias, que se utiliza como contexto para explicar la historia del mundo que se desarrolla en el juego.

Una vez definido el contexto, se diferencian las necesidades de las que requiere la historia principal y se establece el número de historias secundarias por cada variable a estudiar y se desarrolla siguiendo las necesidades de dicha variable.

Para poder analizar las variables, se establece la estructura de las historias, representada en la figura 4, que registra la información necesaria para estudiar las variables. El número de capítulos define la longitud de las historias mientras que las escenas contienen la información de los lugares en los que se desarrollan y, por tanto, por los que pasa el jugador. También contiene la información de los diferentes personajes que están involucrados en dicha escena, por lo que también se utiliza para el cálculo de la cantidad de personajes involucrados.

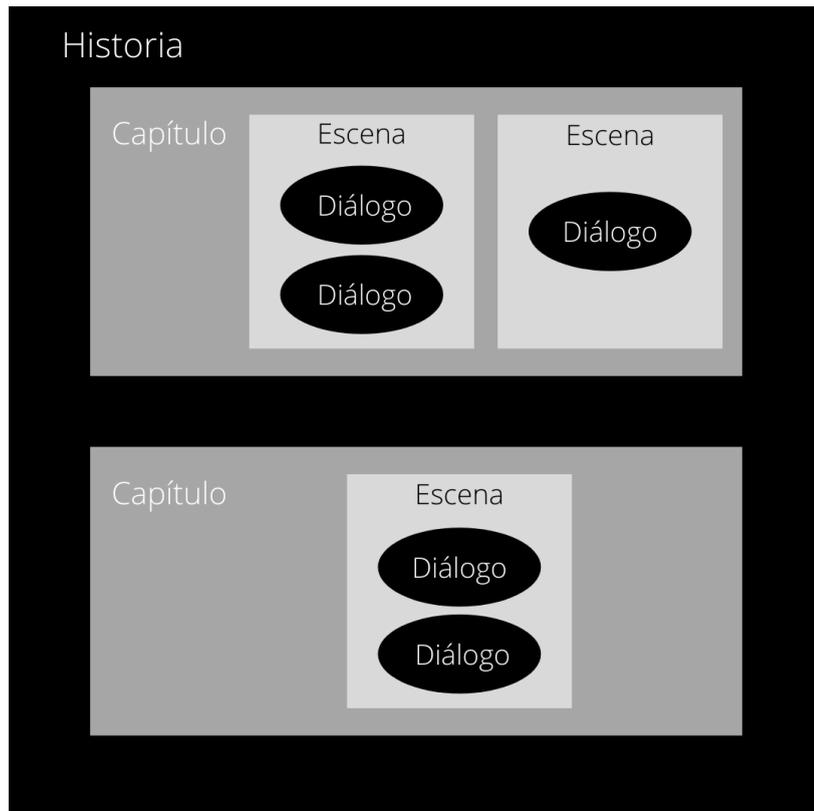


Figura 4: Describe el diseño de la composición de una historia, formada por una serie de capítulos definidos por un conjunto de escenas. Estas escenas desencadenan una serie de diálogos al llegar al lugar indicado o al interactuar con el personaje correcto.

3.1.1 Contexto general de las historias

A la hora de desarrollar las diferentes tramas del juego, es necesario crear un contexto que permita relacionar las historias en un contexto narrativo que facilite el diseño de historias y que permite crearlas de forma que puedan adaptarse a las necesidades del proyecto.

Uno de los contextos más comunes en los videojuegos y que ofrecen diferentes formas de desarrollo de la trama es aquel en el que se encuentra un grupo definido como “los héroes” del mundo que deben enfrentarse a un grupo de villanos. Por ello, las tramas del juego giran en torno a dos organizaciones que se encuentran en constante conflicto: el grupo de “Los Conquistadores” y el grupo de “Los Salvadores”.

El primer grupo cuenta con una jerarquía interna definida por un rango que se establece según el número de territorios que controlan sus miembros, por lo que el objetivo de sus integrantes es conquistar las ciudades para poder crecer en su comunidad.

Por otro lado, el segundo grupo y al que pertenece el protagonista, debe liberar a las ciudades del control del grupo rival para obtener mayor prestigio en el mundo y en su propia comunidad, pudiendo alcanzar el título de héroe. También obtiene prestigio ayudando a cualquier persona que lo necesite.

Los Conquistadores que gobiernan en las ciudades, cuentan con unos objetos de poder que se alimentan del propio control que tienen sobre un territorio. Estos objetos transfieren parte de su poder al líder de la comunidad, volviéndose cada vez más poderoso a medida que sus subordinados conquistan territorios, convirtiéndose en gobernador de todos los territorios conquistados.

3.1.2 Desarrollo de las historias

Con el contexto definido, se analiza la situación narrativa concreta necesaria para desarrollar las diferentes historias del juego. En una versión completa, el juego cuenta con tres grandes regiones con sus gobernadores y sus objetos de poder, con una zona central en la que derrotar al líder de la organización.

Sin embargo, para el desarrollo del proyecto solo es necesario el desarrollo de la primera región del juego en la que desarrollar una parte mínima y necesaria de la trama principal y un total de tres historias secundarias.

Historia principal

La hipótesis establecida es la que define las características que cumple esta historia. Por ello, se desarrolla manteniendo unos valores medios en sus variables que superan a los definidos en las historias secundarias.

El jugador comienza la aventura en su hogar ubicado en Fyrst, el primer pueblo de la región. Al salir a la plaza, puede iniciar una serie de diálogos con un mensajero que le comunica que el alcalde requiere de su presencia. En caso de aceptar dicha misión, el jugador recibe indicaciones para llegar hasta el ayuntamiento, el edificio más grande del

pueblo, donde el alcalde requiere de los servicios del protagonista para obtener un objeto muy valioso en Dara, el pueblo vecino donde reside un estimado amigo suyo.

Tras viajar por la ruta que une ambos pueblos y llegar a Dara, el protagonista encuentra la casa del amigo del alcalde, donde se encuentra con el cadáver de este y el objeto poderoso junto a él. El protagonista inicia una serie de diálogos que reflejan su necesidad de huir con el objeto al bosque y escapar de esa desagradable escena. Allí se encuentra a una anciana que sabe del poder del objeto y anima al jugador a dirigirse a través de las montañas hacia la capital.

El desarrollo de la trama lleva al jugador a desplazarse por la gran parte del escenario de juego, recorriendo un total de 8 lugares de los 14 planteados cumpliendo así con uno de los requisitos de la hipótesis. En cuanto a la longitud de la historia y la cantidad de personajes involucrados en ella, se desarrolla en 3 capítulos formados por 7 escenas en total en las que intervienen 3 personajes.

Las cifras de las historias secundarias se aproximan a las diseñadas en la historia principal, superando estos valores en algunos casos para analizar la influencia de cada variable definida en la hipótesis.

Secundarias

A diferencia de la historia principal, las historias secundarias se diseñan para analizar de forma individual la influencia de cada una de las variables, por lo que cada una de ellas contará con más cantidad de capítulos, mayor intervención de personajes y mayor número de lugares a visitar respectivamente.

En cuanto al resto de valores que no destacan en esa historia, no deben superar a la cifra establecida en la historia que se centra en analizar la relevancia de esa variable, por ello, una historia secundaria tiene una de sus variables con el valor más alto mientras que el resto de variables no superan los valores máximos que establecen las historias secundarias encargadas de analizar dichas variables individualmente.

La narrativa de estas historias está diseñada para no ser fácilmente reconocibles por el jugador como historias secundarias. Esto permite que puedan ser clasificadas como

historias principales por el jugador y obtener así un mejor análisis de las implicaciones de las variables en su clasificación.

Historia secundaria 1: Longitud de la historia

La primera de las historias secundarias diseñadas se centra analizar la variable de la longitud de capítulo, por lo que cuenta con una cantidad de capítulos similar a la historia principal. Para poder centrarse en esta variable, la intervención de personajes y los lugares que se recorren en esta historia es menor al resto de historias.

Esta historia comienza en el ayuntamiento de Fyrst, donde un anciano cuenta con preocupación que su nieta ha desaparecido en una de las cuevas que hay en la ruta que une Fyrst y Dara. La joven quiere convertirse en una heroína y, por ello, decide viajar a las cuevas para hacerse con una de las espadas legendarias perdidas que serían capaces de acabar con gran facilidad con los conquistadores.

El anciano pide al jugador que vaya en su búsqueda, por lo que es necesario dirigirse a la cueva siguiendo el rastro que ha dejado la joven. Mientras se avanza por el mapa hasta llegar a la cueva y futuramente a la mazmorra conectada a esta, se desencadenan diferentes capítulos en los que el protagonista analiza las pistas que encuentra en su recorrido y que indican la cercanía con la joven.

Tras recorrer la mazmorra y encontrar todas las pistas, la joven aparece en la sala más profunda del complejo sin haber encontrado la espada y con intención de volver a su casa en Dara.

Al llegar, aparecen en escena algunos miembros de su familia muy preocupados por el estado de la joven, desencadenando una breve conversación corta con ellos que concluye con estos ofreciendo su ayuda al protagonista en caso de que lo necesite.

La historia cuenta con 3 capítulos, al igual que la historia principal, formados en conjunto por 6 escenas. Este valor de capítulos es la cifra máxima que el resto de historias secundarias no pueden alcanzar ya que esta historia estudia la longitud en capítulos y, por lo tanto, es la secundaria que más capítulos debe tener. Por ello, la cantidad de personajes y lugares recorridos es menor que el resto de historias secundarias diseñadas.

Historia secundaria 2: Cantidad de personajes involucrados

La variable que se estudia en esta historia se centra en analizar la relevancia que tiene en una historia la cantidad de personajes involucrados en su trama para su clasificación. El diseño de esta lleva al jugador a recorrer los diferentes escenarios del mapa en los que se encuentran diferentes personajes con información necesaria para avanzar en el desarrollo de la historia y completar la misión.

Comienza en la plaza de Fyrst, donde una mujer necesita a un grupo de personas para encontrar un objeto en el bosque. Este grupo de personas se encuentra en la taberna de Dara, el lugar al que tiene que llegar el personaje y donde puede entablar conversación con los personajes del lugar para poder reclutarlos.

Al interactuar con todos estos personajes y en el caso de que el desarrollo de las conversaciones que ofrece cada uno se resuelve correctamente, se unen al grupo permitiendo poder abandonar la taberna y dirigirse al bosque.

Allí se registra la zona en profundidad y se localiza el objeto deseado, momento en el que la historia queda finalizada con un final abierto para una posible continuación de esta usando el objeto como conductor de la trama.

La cantidad de personajes involucrados en esta historia, al ser la encargada de analizar la influencia en los personajes y por lo tanto la que tiene un mayor número respecto a las otras historias secundarias, cuenta con un total de 9 personajes. El resto de sus cifras se acercan a las establecidas en la historia principal, con un total de 3 capítulos formados por 6 escenas en total que llevan al personaje a desplazarse por 9 ubicaciones diferentes.

Historia secundaria 3: Cantidad de lugares recorridos

La última de las historias secundarias se centra en recorrer un gran número de ubicaciones para analizar así la última de las variables planteadas en la hipótesis. El desarrollo de la trama lleva a tratar de localizar a un personaje perdido siguiendo las pistas que se encuentran por la región y que también ofrecen algunas conversaciones con personajes involucrados.

El motivo de la búsqueda se da al hablar con un niño en la plaza de Fyrst, que espera nervioso la llegada de su padre que tarda en regresar de Dara, el pueblo vecino donde el

padre vende sus herramientas fabricadas por él mismo. El niño pide ayuda al protagonista para dar con él, por lo que el jugador sigue los pasos del padre para dar con este.

Al acceder a la taberna de Dara, el tabernero informa al jugador de la ausencia del padre del niño debida a la necesidad de ir a por recursos a la cueva. Allí se encuentra una serie de pistas y objetos interactivables que indican que el padre estuvo en el lugar y atravesó la mazmorra.

En la zona final de la mazmorra, junto a la conexión con Dara, se encuentra una anciana que indica la marcha de un hombre nervioso con dirección al bosque. Si se sigue esa misma dirección, se encuentra el cadáver del padre agarrando un libro en el que hay escrita una carta de despedida hacia el hijo. Si se lee la carta, el padre cuenta que era perseguido por una banda que busca destruir algo que él tenía en su posesión.

El final de la historia se alcanza al llegar a Fyrst para contarle al niño los diferentes sucesos ocurridos y entregarle la carta.

Esta historia analiza la relevancia de los lugares recorridos en una historia, por lo que cuenta con una cifra máxima de 10 lugares diferentes de los 14 diseñados. En cuanto al resto de valores, menores al del resto de historias secundarias, son de un capítulo formado por 5 escenas en las que intervienen en conjunto un total de 4 personajes.

3.2 Diseño del prototipo de juego

Una vez diseñadas las historias y analizados tanto los componentes que las forman como los elementos que hacen avanzar la trama, se utiliza este diseño para estructurar las necesidades del prototipo del juego. Este diseño comienza en las primeras fases del proyecto, donde se define el estilo y los sistemas del juego desarrollado en Unity y del que se hace uso en las siguientes fases para poder probar el motor de procesado de narrativa.

De forma paralela, se diseña por un lado el estilo gráfico del juego por el que el jugador puede interactuar y desplazarse y, por otro lado, los sistemas que gestionan el avance en las historias y la interacción del jugador con elementos del juego para poder utilizar el juego correctamente.

Para mejorar la experiencia del juego y permitir que este fuese visualmente más dinámico, también se diseña un sistema en el que los personajes se van desplazando por el mapa, de forma que cada vez que transcurre un intervalo de tiempo definido, el personaje se desplaza una distancia y en una dirección aleatorias, desplazándose en todos los casos a una velocidad constante. En caso de colisionar con algún elemento del juego, el personaje cambia de dirección hasta poder desplazarse, mientras que si el jugador interacciona con dicho personaje, este se detiene.

El conflicto que ocasiona este sistema es la dificultad que podría suponer al jugador para encontrar a los personajes en caso de que estos se hubieran distanciado mucho de su origen, por lo que a pesar de ser diseñado, el sistema no es utilizado en el proyecto.

3.2.1 Estilo del juego

El juego es un de rol o RPG en 2D ambientado en fantasía medieval en el que se encuentran ciudades pequeñas donde se desarrollarán las diferentes historias con la intervención de algunos personajes no jugables.

La trama principal conduce a completar diferentes misiones en diferentes ubicaciones para obtener un objeto clave que cierra la historia de ese lugar y te permite después acceder a la siguiente ubicación disponible.

Para el estudio del proyecto, solo es necesaria una región grande en la que se obtenga uno de esos objetos, por lo que se diseña tanto el mapa como las historias y elementos involucrados en ellas desarrolladas en la primera ubicación disponible del juego.

En cuanto a las tramas secundarias, aunque su trama esté alejada de la trama principal, utilizarán una dinámica parecida para ser completadas aunque no desbloqueen zonas nuevas de juego.

La dinámica del juego consiste en entablar conversación con todos los personajes localizados en el mapa para ir progresando en las diferentes historias que se ofrecen. Para ello el juego necesita la intervención de diferentes sistemas que gestionen el progreso de las historias, el movimiento del propio personaje para poder desplazarse por el mapa y la interacción de dicho personaje con los elementos del juego.

3.2.2 Escenario

Durante el diseño de las historias se analizan las necesidades de las que se requiere para cumplir con la hipótesis y estudiar cada una de las variables que la forman. Para ello, se necesita de un escenario que ofrezca un espacio suficiente para analizar la variable de que estudia la cantidad de lugares recorridos y que permite diferentes opciones para desarrollar la trama de las historias.

Para ello, se diseña en primer lugar un boceto inicial de los lugares y las conexiones entre estos que forman el escenario sobre el que se desarrolla el juego, como se puede ver en la figura 5.

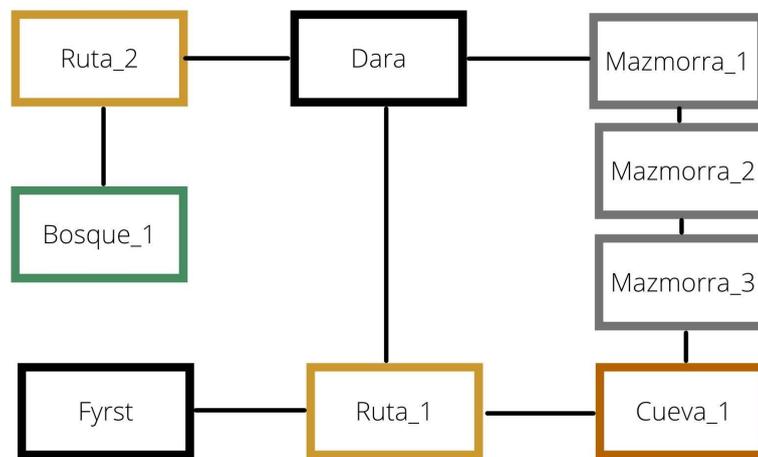


Figura 5: Boceto del escenario de juego con los diferentes lugares y conexiones entre estos. El pueblo donde comienza la historia y donde existe la posibilidad de iniciar las historias secundarias es "Fyrst". Desde allí se abre una ruta que da acceso a una zona de cuevas y a la siguiente ciudad donde se desarrollan las tramas diseñadas. Este pueblo cuenta además con un acceso a una ruta que conecta con el bosque donde finaliza la historia principal.

Una vez definido el boceto, se analiza cada una de las ubicaciones para estudiar las ubicaciones internas que puedan necesitar para ser coherentes con los diseños de las historias.

En su primer diseño, los pueblos cuentan con algunas ubicaciones internas más en formato de casa, tienda o taberna, pero dadas las necesidades reales de las diseñadas, muchas de esas ubicaciones desaparecen y se convierten en elementos decorativos de los pueblos. El diseño del exterior del mapa final se puede ver en la figura 6.



Figura 6: Mapa final implementado en Unity siguiendo el esquema del boceto y las ubicaciones internas necesarias de cada región.

En cuanto a las ubicaciones internas diseñadas para el juego, se encuentran distribuidas entre los dos pueblos. En Fyrst se puede acceder a la casa del protagonista y al ayuntamiento que corresponden a las figuras 7 y 8, mientras que Dara cuenta con acceso a una casa y a la taberna que corresponden a las figuras 9 y 10.

- Ubicaciones de Fyrst (figs. 7, 8):



Figura 7: Casa del protagonista ubicada al noreste de la plaza de Fyrst. En esta ubicación comienza el juego y la historia principal.



Figura 8: Ayuntamiento de Fyrst. Corresponde con el edificio más grande situado junto a la casa del protagonista.

- Ubicaciones de Dara (figs. 9, 10):



Figura 9: Casa ubicada al noreste de Dara. La casa interviene en la historia principal como la casa del amigo del alcalde y en la primera historia secundaria, hogar de la familia de la joven desaparecida.



Figura 10: Taberna de Dara. Es el edificio más grande situado en el centro del pueblo. La segunda y la tercera de las historias secundarias llevan al protagonista a interactuar con esta ubicación.

3.2.3 Sistemas del prototipo de juego

En cuanto a los sistemas del juego, se centran en dos aspectos concretos que abarcan por un lado la carga de las historias y el registro tanto del estado de estas como del punto narrativo en el que se encuentra cada una de ellas a medida que el jugador avanza sus tramas.

En la carga y progresión de las historias intervienen los sistemas de gestión de misiones, que solicita al motor la información de cada historia y controla su estado, y el sistema de diálogos, un sistema creado para que el jugador pueda interactuar, comprender y avanzar en la historia.

Para que estos sistemas puedan progresar, es necesaria la intervención del sistema de interacción y movimiento, que detecta las acciones del jugador con el entorno del juego y permite a los otros sistemas ejecutarse al interactuar con elementos que hacen avanzar la trama.

Sistema de gestión de misiones

A medida que el jugador inicia las diferentes historias de las que dispone, el juego tendrá que ser capaz de gestionar el progreso de los capítulos que forman una historia.

Antes de iniciar la historia, el sistema necesita la información para poder iniciarla, por lo que es necesario establecer el punto concreto en el que el sistema carga la información necesaria para desarrollar las tramas. Para cargar esta información se identifican dos tipos de carga dependiendo del tipo de historia que necesite el juego, la carga de la historia principal y la carga de las historias secundarias.

La historia principal, al ser la primera en ejecutarse, es cargada en el juego desde que este es iniciado, mientras que las historias secundarias se cargan por la ubicación en la que se inicie su desarrollo, buscando entre las secundarias disponibles las que inicien su capítulo en un lugar específico, evitando así cargar todas las historias de una vez y solicitándolas a medida que el jugador avance en el mapa.

En cuanto al inicio de las historias, la principal se inicia al arrancar el juego mientras que para iniciar las secundarias es necesario encontrar al personaje en el mapa que inicie dicha historia.

A medida que el personaje interactúa con los diferentes personajes que hacen progresar la historia, el juego registra los cambios de capítulo de la historia que se está desarrollando, finalizando dichos capítulos una vez que el personaje interactúa con el último personaje del capítulo.

Sistema de gestión de diálogos

Las historias necesitan poder avanzar de forma que el jugador pueda entender e interactuar con la trama, por lo que el juego necesita el sistema de diálogos que controla la carga de los diálogos y, en segundo lugar, la ejecución de estos mientras los muestra al jugador. Tanto para la carga como para la ejecución, el sistema trabaja de dos formas diferentes dependiendo de si nos encontramos con un diálogo de desarrollo de historia o un diálogo genérico que no cuenta como parte de las historias y pueden ejecutarse varias veces.

Los diálogos genéricos se cargan cuando el jugador se encuentra en el mismo lugar que el personaje que ejecuta dicho diálogo, por lo que el sistema solicita los diálogos al cargar los lugares y los personajes ubicados en estos. Como estos diálogos no siguen una progresión dentro de las historias, se pueden ejecutar repetidas veces.

A diferencia de los diálogos genéricos, los diálogos destinados al progreso de la historia solo se cargan cuando el jugador se encuentra en el punto de la historia en el que se requiere ese diálogo, por lo que al cargar los capítulos de las historias se cargan también estos diálogos.

Este tipo de diálogos también se diferencian de los anteriores en que no son repetibles, ya que la historia va avanzando y no permite dar saltos hacia atrás en esta. Pero para dar más libertad al jugador y favorecer que la historia principal y las historias secundarias no sean fácilmente reconocibles, se diseñan los primeros diálogos para que las historias puedan ser rechazadas y, al no iniciarse la historia, permite volver a ejecutar el diálogo.

Además, este tipo de diálogos cuentan con diferentes posibilidades de progresión e inicialización, como aquellos diálogos que ofrecen diferentes opciones de selección que desencadenan diferentes respuestas de los personajes o diálogos de pensamiento del personaje que el sistema inicia una vez que el jugador entra en una zona concreta del mapa o interactúa con un objeto involucrado en la trama.

Sistema de control e interacción

Una vez definidos los sistemas que hacen avanzar las historias, es necesario definir un sistema que permita al jugador activar dichos sistemas. En este punto interactúa el sistema de interacción del personaje, tanto con los personajes y objetos como con el resto de elementos del entorno, permitiendo así lanzar en cada interacción la llamada necesaria al sistema que el juego necesite en ese momento para mostrar, gestionar y desarrollar la historia.

Este sistema controlará las interacciones del jugador con el entorno por medio de una tecla establecida para ese tipo de función, de forma que el jugador al estar a cierta distancia de un elemento interactivo del juego y presionar dicha tecla, el sistema identifica el tipo de elemento con el que se interactúa e inicia la funcionalidad concreta asociada a dicho

elemento. Pero para que el jugador pueda alcanzar esa distancia necesita de un sistema que le permita desplazarse por el mapa.

El sistema de movimiento es el que permite al jugador desplazar a su personaje con la intervención de una serie de teclas definidas para controlar el movimiento que, al igual que la interacción, ejecutará su acción al presionar alguna de dichas teclas.

En relación con el desarrollo del juego y el resto de sistemas, se diferencian dos tipos de interacción, utilizando en uno de estos casos el sistema de movimiento.

Para que las historias se desarrollen correctamente, el juego necesita establecer una interacción entre el jugador y los personajes repartidos por el mapa. Este tipo de interacción ejecuta el diálogo asociado al personaje cumpliendo con las condiciones dadas en el sistema de diálogos y gestión de historias.

Además, en el caso de los diálogos que intervienen en dichas historias, también intervienen ambos sistemas descritos, utilizando el sistema de interacción para poder avanzar entre los diferentes diálogos y seleccionar una respuesta en caso de que el diálogo sea un diálogo de selección de opciones, y el sistema de movimiento para poder desplazarse entre las diferentes opciones que ofrece este tipo de diálogo .

El otro tipo de interacción se establece entre el jugador y los accesos a zonas cerradas de algunas zonas del mapa, empleando la interacción para permitir al jugador salir o entrar de estas zonas y poder así cambiar de lugar.

Capítulo 4: Diseño computacional del motor de procesamiento de historias

Antes de poder iniciar el juego, es necesario cargar antes las historias clasificadas para que el sistema que controla el juego las gestione según su tipo. El motor de procesamiento de narrativas es el sistema independiente al prototipo del juego que clasifica dichas historias y las envía al juego en el punto concreto en el que este la necesite.

Las primeras fases de este sistema se centran en crear una primera versión del motor capaz de cargar la información de las historias a partir de una serie de archivos que almacenan la información de las historias y del entorno en el que se desarrolla el juego.

Una vez que la hipótesis es aceptada o rechazada, la segunda fase del motor analiza las variables propuestas en la hipótesis y clasifica las historias según la media de las cifras obtenidas, siendo la historia principal la que mayor valor obtenga.

4.1 Arquitectura UML

Al definir cada sistema y cada estructura de la información a representar, se genera un diagrama UML que hace un planteamiento más profundo del diseño del que surge la implementación desarrollada en el **Capítulo 5: Implementación**. Las figuras 11 a 15 representan las diferentes clases del diagrama, la figura 16 representa el diagrama completo para poder visualizar las relaciones entre clases.

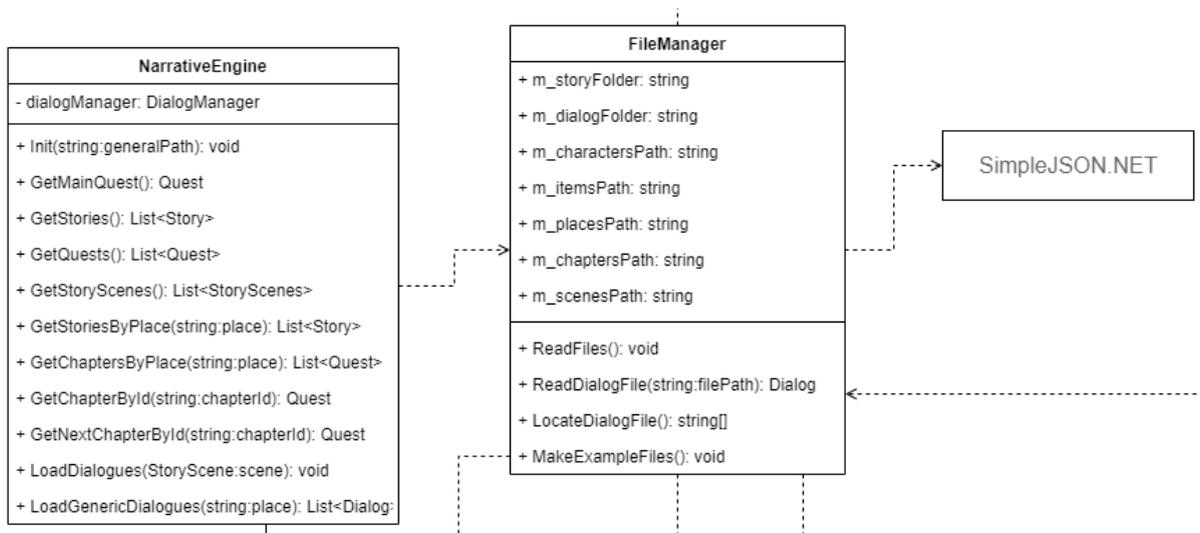


Figura 11: Parte del diagrama de clases del motor, mostrando las clases NarrativeEngine y FileManager y la librería SimpleJSON.NET.

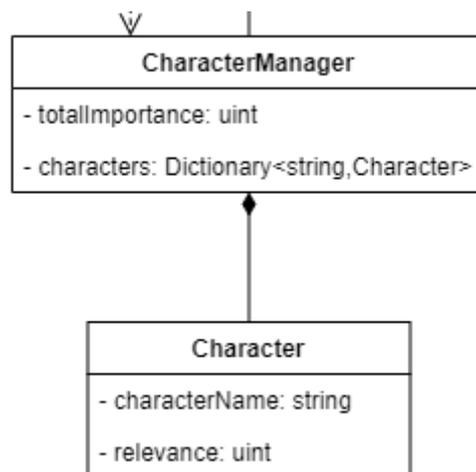


Figura 12: Parte del diagrama de clases del motor, mostrando las clases CharacterManager y Character.

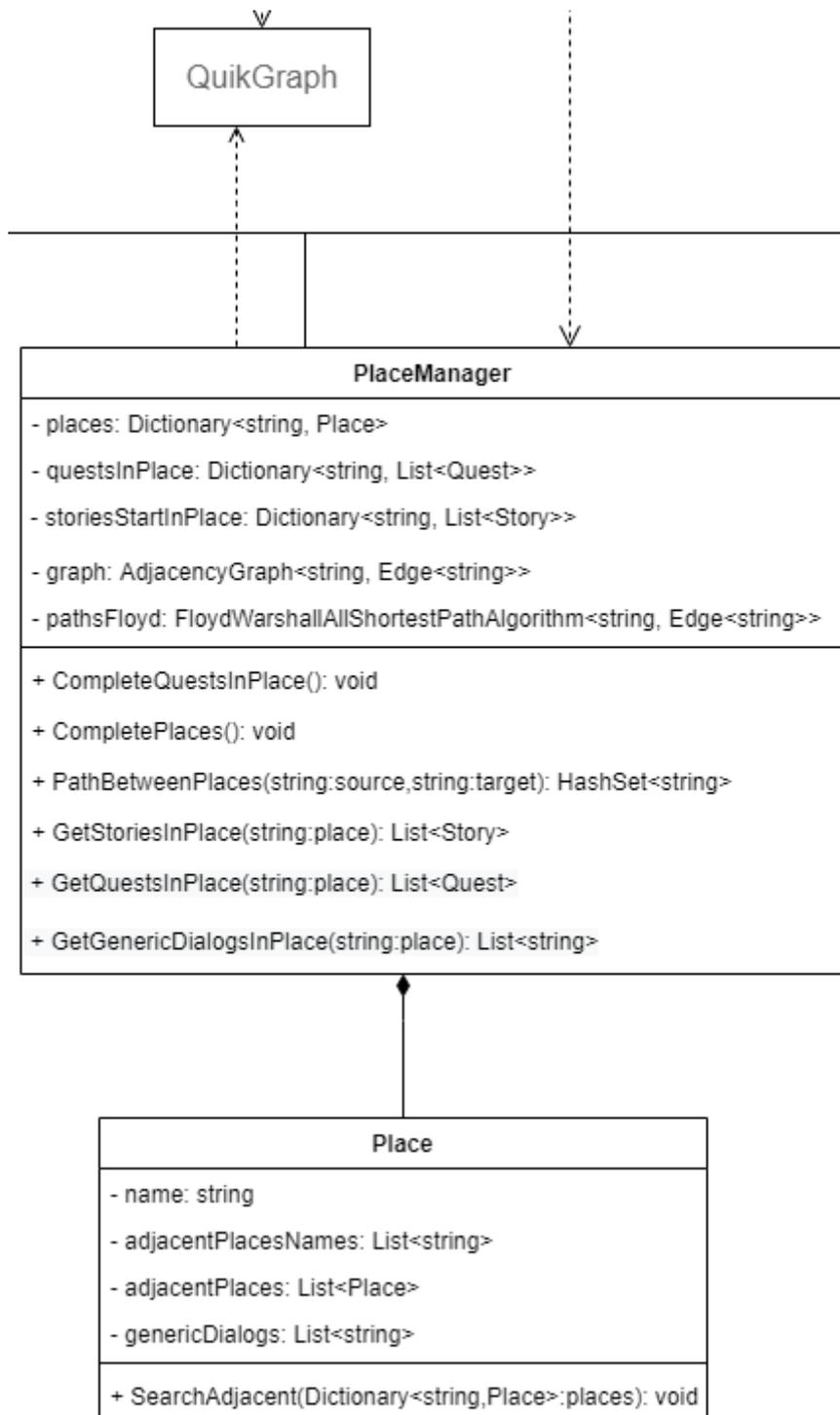


Figura 13: Parte del diagrama de clases del motor, mostrando las clases PlaceManager y Place y la librería QuikGraph.

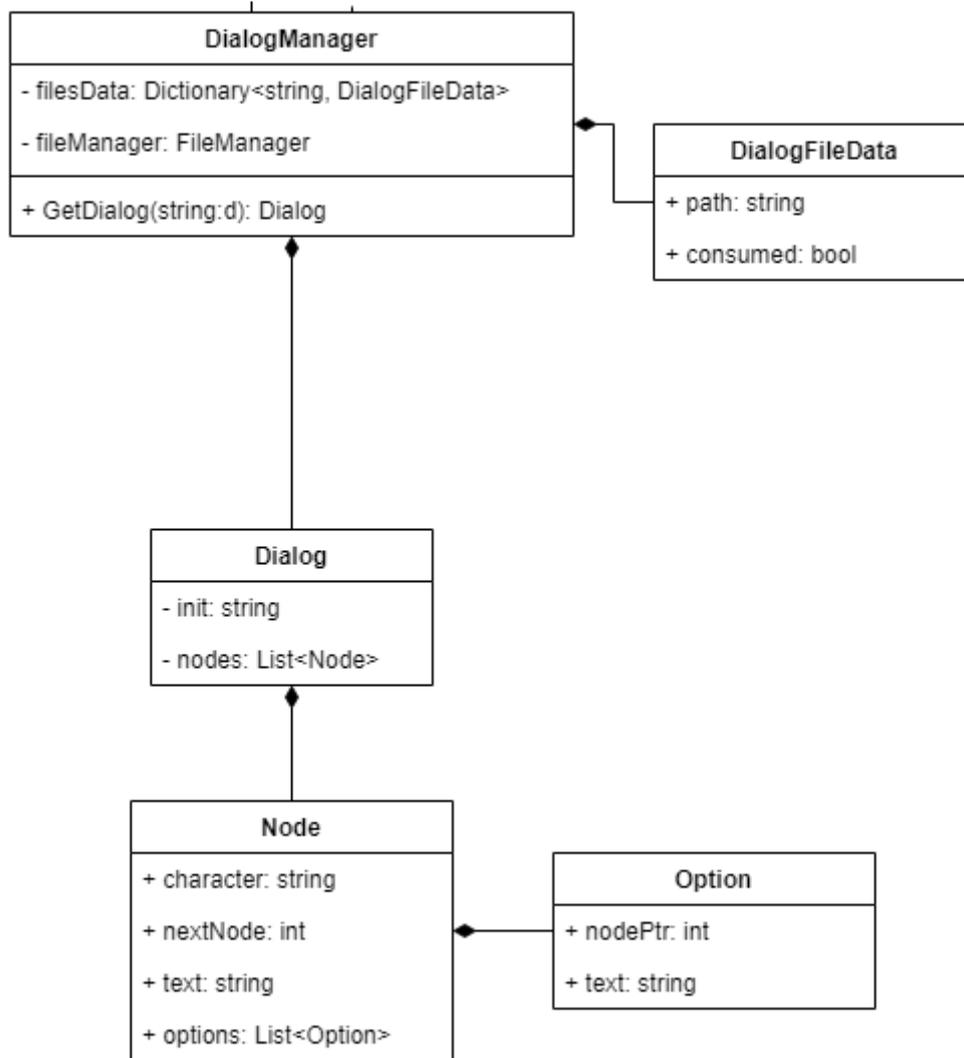


Figura 14: Parte del diagrama de clases del motor, mostrando las clases DialogManager, DialogFileData, Dialog, Node y Option.

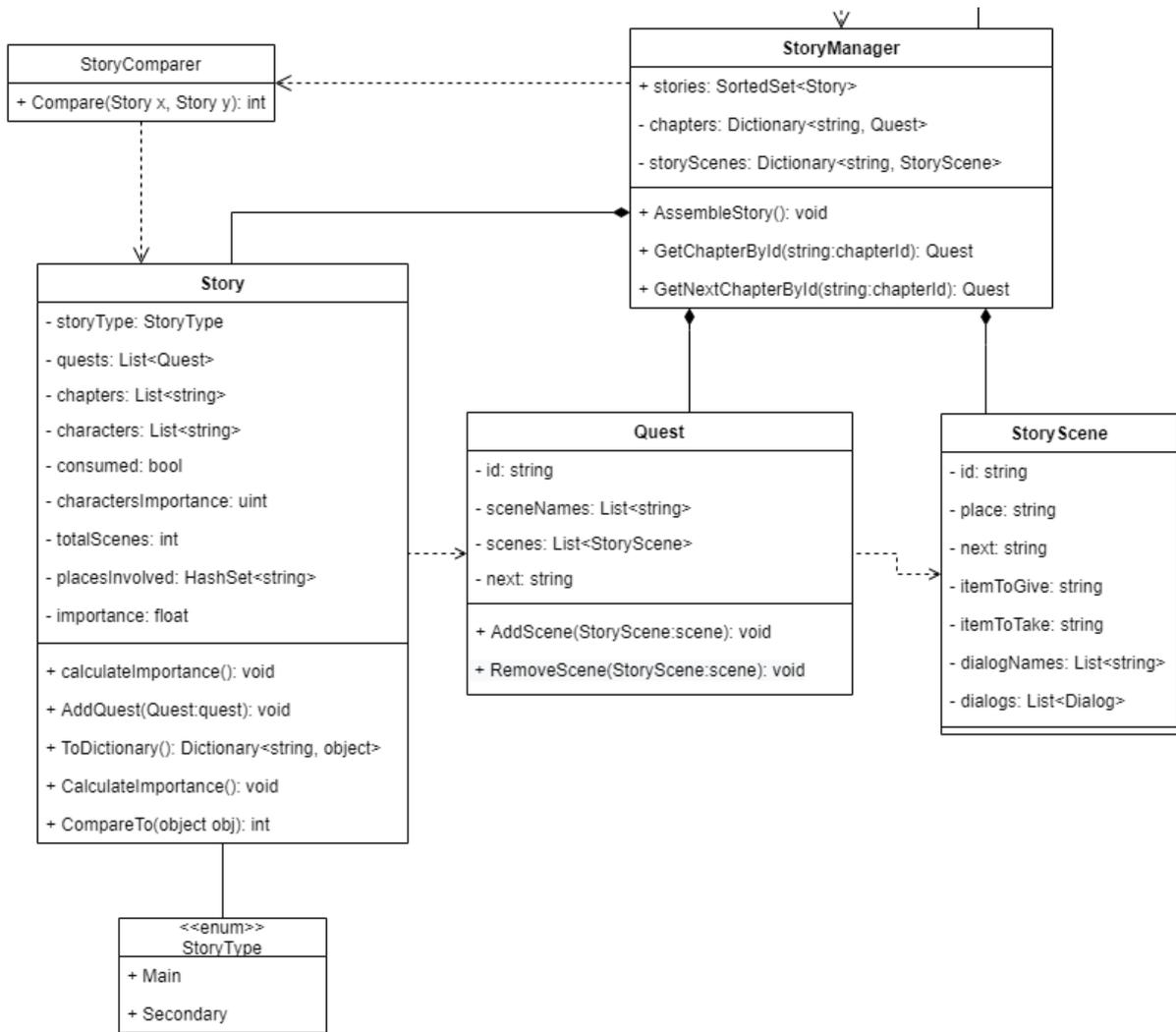


Figura 15: Parte del diagrama de clases del motor, mostrando las clases StoryManager, Story, StoryComparer, StoryType, Quest y StoryScene.

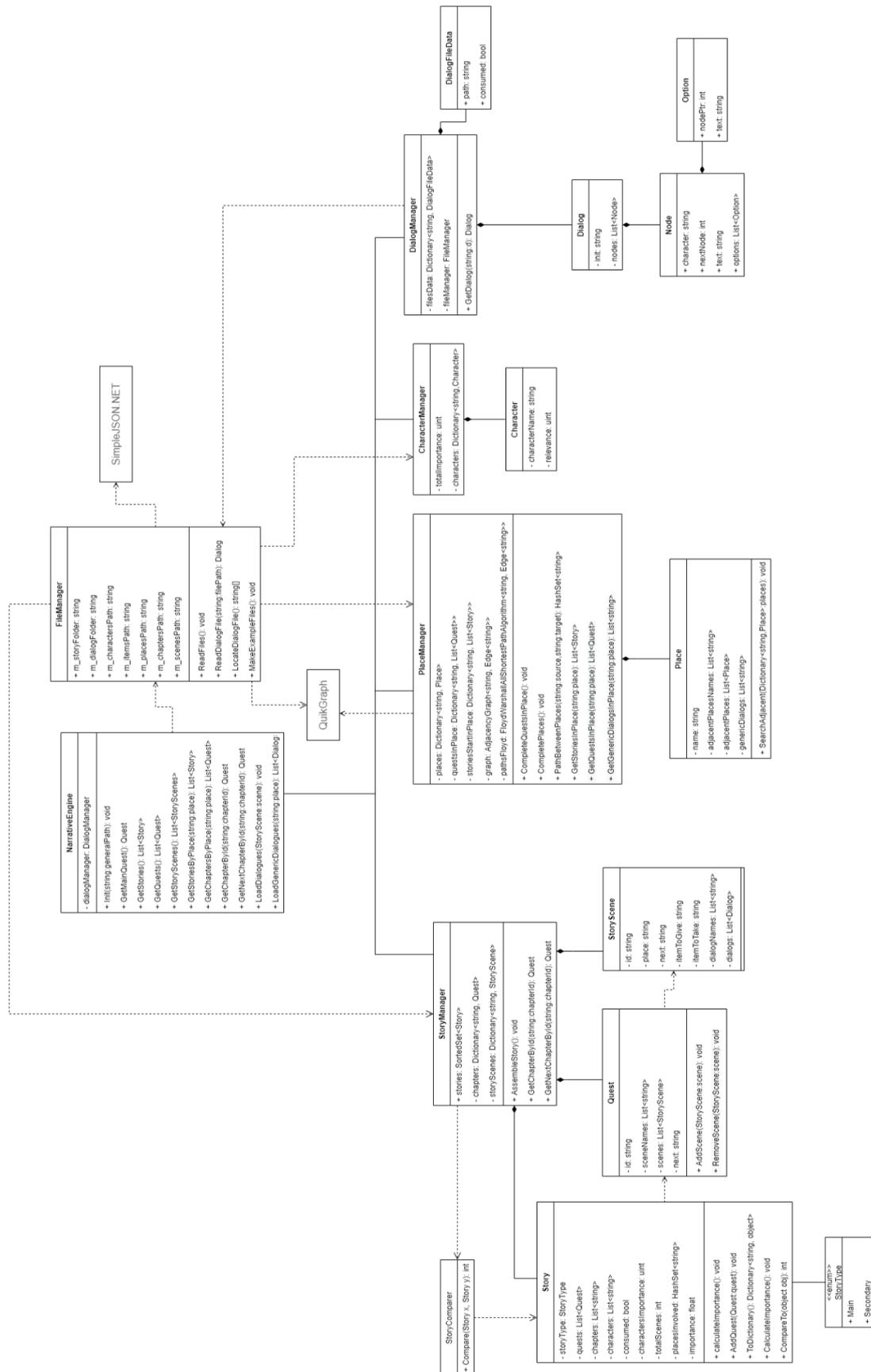


Figura 16: Diagrama de clases del motor.

4.2 Estructura inicial de la información

Una vez definido el escenario de juego y las historias junto a sus diálogos y los personajes involucrados en estos, se traslada toda la información en un formato json para facilitar la lectura en el motor. Cada uno de los elementos que forman la historia se almacenan con esta estructura en un archivo diferente a excepción de los diálogos, que al ser una gran cantidad de diálogos con mucha información cada uno de estos, es mejor crear un archivo por cada diálogo.

En las primeras fases del diseño, se plantea organizar la información de manera que la entidad Mapa estaba formada por una serie de historias a las que se le asignaba unos Estados que registran la localización del personaje en la historia que se desarrollaba. Los estados también contienen una lista de las misiones asociadas a dicho estado.

Al avanzar en el diseño, se estudia la estructura planteada y se rediseñan sus nombres y algunos de sus atributos para que sea mucho más comprensible tanto a la hora de implementar las clases que almacenan la información como para que sea comprensible para otros diseñadores o implementadores.

4.2.1 Estructura de los personajes

Contiene una lista con los nombres de los personajes que existen en el juego, tanto si son personajes involucrados en las historias como si son personajes con diálogos genéricos. De esta forma se facilita la búsqueda de estos (figura 17).

```
{  
    "Personaje1",  
    "Personaje2"  
}
```

Figura 17: Estructura en forma de lista en la que se almacena la información de los personajes.

4.2.2 Estructura de los objetos

Al igual que la estructura de personajes, la estructura del archivo de objetos contiene una lista con los nombres de aquellos objetos relevantes en las historias. Al igual que los personajes, esta estructura permite una búsqueda rápida de los objetos (figura 18).

```
{
    "Objeto1",
    "Objeto2"
}
```

Figura 18: Estructura en forma de lista en la que se almacena la información de los objetos.

4.2.3 Estructura de los lugares y sus conexiones

La información de lugares se define con una serie de objetos formados por el nombre del lugar, que se emplea como identificador de este, y una lista con los nombres de lugares adyacentes a los que se puede acceder desde dicho lugar.

También cuenta con una lista de diálogos genéricos que se pueden iniciar en dicho lugar, separando así las escenas con diálogos que desarrollan la trama de los diálogos genéricos. Como hay lugares con personajes que solo hacen avanzar la trama o que directamente no cuentan con personajes, la lista de diálogos genéricos puede no contener ningún diálogo.

Esta estructura permite conocer la estructura del mapa, ya que cuenta con las conexiones que hay entre las diferentes ubicaciones (figura 19).

```
{
    "Nombre": "Lugar1",
    "Nombre Lugares Adyacentes":
    [
        "Lugar2",
        "Lugar3"
    ],
    "Diálogos genéricos":
    [
        "DiálogoGenérico1",
        "DiálogoGenérico2"
    ]
}
```

Figura 19: Estructura que recoge una serie de objetos con la información de cada lugar.

4.2.4 Estructura de las historias

Las historias cuentan con un identificador de tipo para clasificar las historias en principal y secundarias para la primera versión del motor en la que solo se carga la información y no se clasifican las historias. Los valores puede tener son 0 para la historia principal y 1 para las historias secundarias. También cuenta con una lista con una lista con los identificadores de los capítulos que componen dicha historia.

De esta forma, la identificación de la historia en esta primera versión del motor es directa, además de poder extraer la lista ordenada de los capítulos que forman dicha historia (figura 20).

```
{
  "Tipo Historia": 0,
  "Capítulos":
  [
    "Capítulo1",
    "Capítulo2"
  ],
}
```

Figura 20: Estructura en la que almacena una serie de objetos con la información de cada historia.

4.2.5 Estructura de los capítulos

Los capítulos de las historias contienen un identificador del propio capítulo, que se utiliza en las historias para poder asociarlo, y una lista de identificadores de escenas que componen el capítulo.

También cuenta con el identificador del siguiente capítulo de la historia para poder acceder a este una vez terminado el anterior. En caso de ser el último capítulo de esa historia, ese atributo será un string vacío.

La estructura de este elemento se diseña, al igual que las historias, para ser fácilmente localizables a la hora de que el motor pueda proporcionar la información del capítulo al juego y de forma que obtenga la lista ordenada de las escenas que lo forman (figura 21).

```

{
  "ID": "Capítulo1",
  "Escenas":
  [
    "Escena1",
    "Escena2"
  ],
  "Siguiete Capítulo": "Capítulo2"
}

```

Figura 21: Estructura en la que almacena una serie de objetos con la información de cada capítulo.

4.2.6 Estructura de las escenas

Al igual que los capítulos, las escenas que conforman los capítulos cuentan con un identificador para poder ser asociados con el capítulo al que pertenecen y el identificador de la escena siguiente en caso de que haya. A su vez, las escenas cuentan también con el nombre del lugar concreto en el que se desarrolla la escena, permitiendo así asignar escenas a lugares concretos, unos campos creados para tratar los objetos que puedan dar o solicitar los personajes de dicha escena.

En último lugar, se encuentra una lista con los identificadores de los archivos que contienen la información de los diálogos que suceden en esa historia. En una primera versión de esta estructura los diálogos genéricos eran almacenados en esta lista hasta que durante el desarrollo se trasladaron a lugares donde tenía más coherencia.

También cabe destacar que en las primeras versiones de esta estructura había algunas escenas que no contenían diálogos y se empleaban a modo de transición, pero eso marcaba mucho el recorrido que debe hacer el jugador, por lo que se eliminan algunas escenas y se establece el requisito de que las listas de diálogos nunca pudiesen estar vacías.

El diseño de esta estructura ayuda a encapsular todas las necesidades que pueda tener la trama para completar con un objetivo concreto de esta. Además también ayuda a conectar los diferentes puntos por lo que transcurre la historia, por lo que es más fácil analizar las posibles rutas entre escenas de un capítulo (figura 22).

```

{
  "ID": "Escena1",
  "Lugar": "Lugar1"
  "Siguiete Escena": "Escena2"
  "Objeto A Dar": "Objeto1"
  "Objeto a Tomar": "Objeto2"
  "Diálogos":
  [
    "Diálogo1",
    "Diálogo2"
  ]
}

```

Figura 22: Estructura en la que almacena una serie de objetos con la información de cada capítulo.

4.2.7 Estructura de los diálogos asociados a los personajes

Los archivos con la estructura de los diálogos tienen como nombre el identificador que se utiliza para asociar los diálogos con las escenas donde se desarrollan o los lugares en los que pueden activarse en caso de ser genéricos.

En cuanto a la estructura, contienen el nombre del personaje que inicia la conversación y con el que el jugador debe interactuar para iniciarlo. A continuación, se define una lista de nodos con la información de la conversación, siendo cada uno de estos nodos un objeto que contienen la información del personaje que habla en ese momento concreto, la posición o índice dentro de la lista del siguiente nodo, el texto que se muestra al jugador y, en caso de que tenga opciones de selección, una lista con cada una de dichas opciones. Estas opciones contienen el texto que muestra cada opción y el nodo siguiente al que se llega a través de esa opción.

La estructura descrita permite organizar la información de forma que cada diálogo tenga la información del personaje que lo inicia, manteniendo la conexión entre los diferentes nodos que forman dicho diálogo y permitiendo al diseñador diferentes configuraciones de estos.

En cuanto al campo que almacena la información del siguiente nodo siguiente dentro de un nodo también se emplea para definir diferentes tipos de opciones dentro del propio diálogo:

- Opción numérica: el número que identifica la posición del siguiente nodo dentro de la lista (figura 23).

```

{
  "Inicio": "Personaje1",
  "Nodos":
  [
    {
      "Personaje": "Personaje1",
      "Siguiete Nodo": 1,
      "Texto": "Mensaje del Personaje1."
      "Opciones": []
    },
    {
      "Personaje": "Personaje2",
      "Siguiete Nodo": 2,
      "Texto": "Mensaje del Personaje2."
      "Opciones": []
    }
  ]
}

```

Figura 23: Estructura en la que almacena la información de un diálogo.

- Opción "-1": Se utiliza cuando el diálogo contiene opciones, la información del nodo siguiente la almacena cada una de las diferentes opciones, pudiendo compartir el mismo nodo siguiente o nodos diferentes (figura 24).

```

{
  "Inicio": "Personaje1",
  "Nodos":
  [
    {
      "Personaje": "Personaje1",
      "Siguiete Nodo": -1,
      "Texto": "Mensaje del Personaje1"
      "Opciones":
      [
        {
          "Nodo Objetivo": 1,
          "Texto": "Opción1"
        },
        {
          "Nodo Objetivo": 2,
          "Texto": "Opción2"
        }
      ]
    }
  ],
}

```

Figura 24: Estructura en la que almacena la información de un diálogo con opciones.

- Opción "-2": Indica que se ha finalizado dicho diálogo (figura 25).

```

{
  "Inicio": "Personaje1",
  "Nodos":
  [
    {
      "Personaje": "Personaje1",
      "Siguiete Nodo": -2,
      "Texto": "Mensaje del Personaje1."
      "Opciones": []
    }
  ]
}

```

Figura 25: Estructura en la que almacena la información de un diálogo con finalización.

- Opción "-3": Permite que el diálogo no sea consumido y, por lo tanto, permite repetir la conversación. Esta opción es exclusiva de los diálogos genéricos o de los diálogos que inician misión, aunque puede utilizarse para otros casos (figura 26).

```

{
  "Inicio": "Personaje1",
  "Nodos":
  [
    {
      "Personaje": "Personaje1",
      "Siguiete Nodo": -3,
      "Texto": "Mensaje del Personaje1",
      "Opciones": []
    }
  ]
}

```

Figura 26: Estructura en la que almacena la información de un diálogo genérico.

4.3 Primera versión del motor

Al establecer el diseño de las diferentes estructuras de los elementos, se diseñan tanto las clases, que siguen dichas estructuras, encargadas de almacenar la información como el sistema encargado de cargar toda esa información de los archivos y el resto de sistemas encargados de enviar al juego la información de forma organizada.

4.3.1 Sistema de carga y almacenamiento de información

En el momento en el que el juego inicia se hace una llamada al motor para que localice los archivos de la información e inicie la lectura de estos. Una vez encontrado, asocia cada tipo de información a una estructura concreta para almacenar correctamente la información de los diferentes tipos de elementos definidos.

A medida que se leen los archivos, el sistema crea esos elementos como objetos y los almacena en listas o diccionarios dependiendo de si se necesita una rápida localización de estos, almacenando así toda la información del juego y siendo así de fácil acceso. Cada una de estas listas se almacenan en el gestor encargado de tratar la información de cada lista.

4.3.2 Sistema de peticiones del juego

Una vez cargada toda la información en el sistema, se definen las diferentes funciones que enviarán la información al juego a medida que éste solicite los recursos

necesarios. Para ello, se diseñan tres tipos de peticiones enfocadas en cargar la historia principal, cargar las historias secundarias y cargar los elementos que las forman respectivamente.

En la carga de la historia principal, busca la historia identificada como tal y devuelve el primer capítulo que forman dicha historia de manera que el juego inicia directamente el desarrollo de este capítulo situándose en el juego.

Por otro lado, las historias, al ser cargadas solo cuando el jugador se encuentra en el lugar en el que comienza la trama de dicha historia, necesita de un sistema que organice previamente la información para agilizar la petición. Este sistema se encarga de asociar las historias a un lugar del juego concreto mediante la ubicación de la primera escena que inicia el primer capítulo de dicha historia.

Con las historias asociadas a un lugar, el sistema busca las historias que inician en dicho lugar y envía el primer capítulo de cada una de estas historias para poder ser iniciada por el jugador.

A medida que las historias avanzan, el juego necesita cargar la información que hace continuar la trama, por lo que en el momento en el que un capítulo se finaliza, el juego solicita al motor el próximo capítulo en caso de haber un capítulo que continúe la historia, por lo que el motor cuenta con una función encargada de localizar mediante el identificador de un capítulo el siguiente a este y enviarlo en caso de existir al juego.

El caso de carga de las escenas es similar al de los capítulos, el juego solicita la carga de la escena que inicia el capítulo y posteriormente la carga de la siguiente escena a esta mediante su identificador. Pero al cargar las escenas e iniciarlas, también necesita tener la información de los diálogos que se desarrollan en dicha escena, por lo que el juego solicita los diálogos de una escena al motor y este los localiza según su identificador y se lo envía al juego estructurado de forma que este sea capaz de gestionarlo.

En el caso de los diálogos genéricos, se solicitan en el momento en el que el jugador entra en el lugar en el que se desarrollan dichos diálogos, por lo que en ese momento el juego solicita al motor los diálogos y este los busca según su identificador, enviando posteriormente la información al juego.

4.4 Segunda versión del motor

Tras la experimentación realizada en el **Capítulo 6: Pruebas del prototipo**, la hipótesis de partida es rechazada debido a que la cantidad de personajes no es percibida por los jugadores como un elemento que haga distinción entre los tipos de historia.

Sin embargo, la relevancia narrativa de los personajes y su papel dentro del mundo es reconocida como un elemento clave a la hora de identificar historias, por lo que se plantea la nueva hipótesis y se define para incluir la nueva variable y rechazar la anterior.

Por tanto, la hipótesis planteada previamente se ajusta a las necesidades encontradas tras la experimentación de forma que una historia principal se define como aquella historia cuya media de longitud de capítulos, cantidad de lugares visitados y relevancia narrativa de sus personajes, es mayor que la del resto de historias que se plantean dentro de un juego.

Con este cambio en la forma de identificar las historias, se diseña un nuevo sistema que, según el valor numérico asignado a los personajes basado en la información que ofrece al jugador y el papel que desempeña en el mundo, ofrece a la historia mayor o menor importancia narrativa según la relevancia de los personajes involucrados en ella.

Para analizar el resto de variables, la longitud de la historia viene dada por la lista de capítulos que contiene, pero los jugadores no identifican el momento en el que el capítulo cambia, identifican el cambio de escena según que se desarrollan en lugares diferentes, por lo que la longitud de las historias se analiza empleando lo que el jugador percibe, es decir, el número de escenas en vez de el número de capítulos que forman una historia.

En cuanto a la cantidad de lugares recorridos, es necesario analizar no solo los lugares en los que se desarrollan las escenas, también es necesario analizar el recorrido mínimo que hay entre los lugares que unen cada escena con su siguiente escena, por lo que para poder analizar esta variable se crea un grafo con el mapa del juego y sus conexiones para ser más precisos a la hora de obtener la cantidad de lugares que se recorren en una historia.

Por último, se añade en el motor la funcionalidad de clasificar las historias empleando el estudio de estas variables y enviar la información al juego, por lo que algunas de las funciones encargadas de las peticiones con Unity se modifican para incluir todas las funcionalidades añadidas en el motor.

4.4.1 Clasificación de personajes según su relevancia

Para registrar el valor de cada personaje se reestructura dicho componente para que, a la hora de cargar a los diferentes personajes cargue también este valor, por lo que la nueva estructura queda definida como se representa en la figura 27.

```
{
    {
        "Nombre": "Personaje1",
        "Relevancia": 10,
    },
    {
        "Nombre": "Personaje2",
        "Relevancia": 25,
    },
}
```

Figura 27: Estructura en la que almacena la información de los personajes modificada para contener nombre y relevancia de los personajes.

Con esta estructura, el sistema puede cargar fácilmente la relevancia de los personajes, pero para saber qué personajes pertenecen a cada historia, el sistema tendría que revisar todos los diálogos y buscar a cada personaje involucrado, por lo que para agilizar la búsqueda, la estructura que almacena la información de las historias se modifica para contener a parte de la lista de capítulos, una lista con los personajes involucrados en las diferentes tramas (figura 28).

```

{
  "Capítulos":
  [
    "Capítulo1",
    "Capítulo2",
  ],
  "Personajes":
  [
    "Personaje1",
    "Personaje2",
  ],
}

```

Figura 28: Estructura en la que almacena la información de los personajes modificada para contener nombre y relevancia de los personajes.

Al ser un valor asignado por el propio diseñador y no calculado por el motor, ofrece mayor libertad a la hora de crear su propia escala ajustada a sus necesidades. En este proyecto, el valor se calcula mediante la media ponderada de dos valores que miden por un lado la relevancia narrativa del personaje que interviene en la historia a la que pertenece y, por otro lado, la relevancia diegética, que es la importancia del personaje dentro del mundo al que pertenece, como la posición social o la influencia que ejerce sobre este mundo ficticio.

Ambas escalas se encuentran entre los valores de 0 y 100 y, a la hora de calcular la media, tienen una relevancia del 60% en el caso de la importancia narrativa y un 40% en la relevancia diegética. Esta escala tiene una relevancia menor ya que en la historia tiene mayor foco en la trama que desempeña, que es más valorada por los jugadores al analizar los resultados de la primera experimentación desarrollada en el **Capítulo 6: Pruebas del prototipo**.

4.4.2 Sistema de conexiones entre lugares

Con las primeras pruebas con el motor, el recuento de lugares realizado se basa únicamente en las ubicaciones establecidas en las escenas, pero el jugador interpreta como lugares visitados aquellos en los que ocurren escenas con diálogos y el recorrido entre una escena y la siguiente.

Para analizar correctamente el recorrido mínimo que realiza el jugador dentro del mapa, es necesario emplear una estructura que permita analizar dicho recorrido entre

escenas y capítulos. Por ello, la estructura empleada para analizar dicho recorrido es la de un grafo no dirigido empleado para calcular el camino mínimo entre sus vértices, siendo estos los lugares concretos y las aristas las conexiones entre estos. Por ello, en el momento en el que el sistema carga los lugares identificados por su nombre junto a una lista de lugares adyacentes que conectan con estos, el sistema genera dicha estructura. En la figura 29 se puede ver una representación gráfica del grafo formado por los lugares del prototipo de juego.

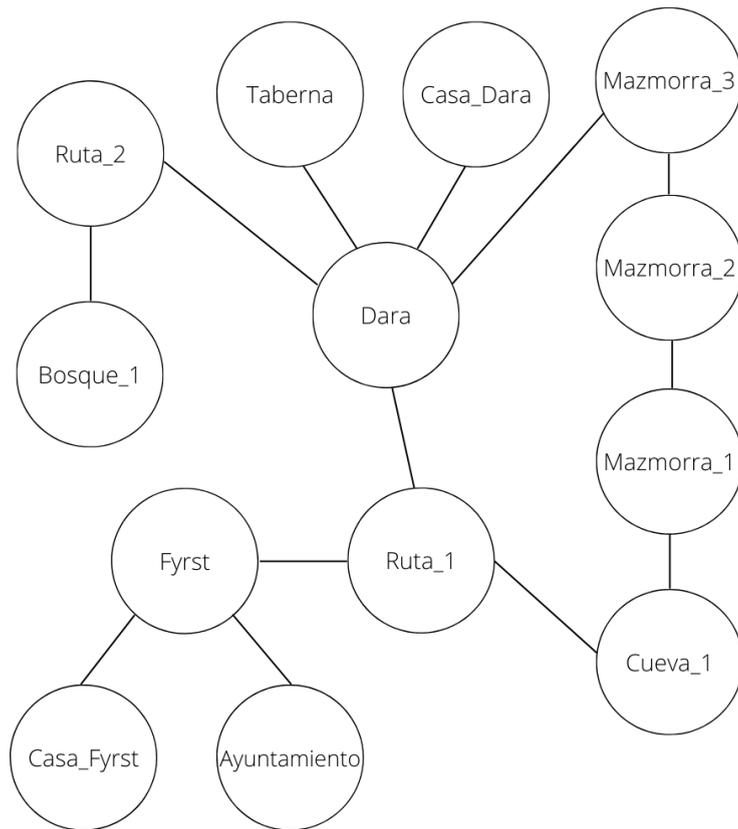


Figura 29: Grafo no dirigido con las conexiones de cada lugar, tanto internas como externas

Después de generar el grafo y cuando el sistema carga los capítulos de las historias, el motor emplea la estructura creada para calcular la distancia mínima entre los lugares en los que ocurren las escenas.

Para calcular esta distancia mínima, se plantea en primer lugar la búsqueda en anchura o BFS, pero este algoritmo solo calcula la distancia más corta para llegar a dicho vértice sin informarte de los vértices por los que ha pasado para calcular la ruta, pudiendo así repetirse lugares que ya han sido recorridos.

Como segunda opción, se estudia la posibilidad de utilizar el algoritmo de Dijkstra debido al coste similar que tiene BFS, pero este algoritmo plantea el problema de que varias historias pueden repetir el mismo camino, duplicando la ejecución de una información que ya conocemos y por lo tanto, ampliando el tiempo de ejecución.

Al final, se estudia el recorrido mínimo entre vértices mediante el algoritmo de Floyd-Warshall, que calcula los caminos mínimos entre dos vértices dados, tomando como valor 1 en sus aristas y mostrando el recorrido seleccionado por este.

Mediante el archivo que almacena la información de lugares se construye el grafo, siendo sus vértices cada lugar ubicación del juego y las aristas las relaciones entre estos. De esta forma, en el momento en el que se cargan las historias, se calcula la cantidad de lugares recorridos en cada una de ellas, almacenando este valor para la posterior clasificación.

4.4.3 Sistema de clasificación de historias

La primera versión del sistema ordenaba las historias en tres listas, una por cada variable a estudiar y analizando las que obtuvieron las primeras posición en estas listas, pero existía la posibilidad a la hora de ordenar las listas de que los valores de las tres variables se repitiera, no obteniendo así una ordenación exacta de estas, por lo que en vez de organizar las historias diferentes listas, se ordenan en una única lista ordenada según la relevancia de historia calculada a partir de las tres variables que componen dicha historia, siendo la de primera posición la historia principal.

Durante el proceso de carga de la información y al ser cargadas las historias, que son el último elemento en cargarse, el sistema clasifica dichas historias, por lo que lo último que hace la función de carga es asignar su tipo dependiendo de las variables que estudia la hipótesis.

En primer lugar, se analiza la nueva hipótesis para estudiar la forma de analizar las historias. En cuanto a la longitud de la historia, el cambio de un capítulo a otro no es percibida por los jugadores, pero sí los cambios de escena, por lo que la longitud de la historia se analiza a través del número total de escenas que forman la historia. Continuando con la siguiente variable, la cantidad de personajes es sustituida por la relevancia narrativa

de estos, obteniendo el valor de la relevancia total de la historia a través de la suma de las relevancias narrativas de sus personajes que está comprendida entre los valores 0 y 100. Por último, la cantidad de lugares incluye la estructura de un grafo que permite calcular la distancia entre las diferentes escenas sin repetir lugares.

Para obtener estos valores, el sistema inicia los cálculos necesarios en el proceso de carga de ficheros. Mientras el sistema se encarga de leer las escenas, también utiliza el grafo ya cargado para calcular la distancia entre los lugares, guardando el valor en una variable para poder utilizarla en la operación de clasificación. Por otro lado, mientras el sistema se encarga de leer la información de las historias, identifica a los personajes involucrados en estas almacenados en la lista de personajes de la historia, busca su relevancia en la lista de personajes ya cargados y suma sus relevancias, almacenando esta información en otra variable. Por último, cuenta el número de escenas que contiene, guardando el valor en otra variable.

Una vez cargados los tres valores necesarios para clasificar las historias, se ejecuta una media ponderada con los tres valores para asignarle a cada una de las historias un nivel de relevancia.

Tras los resultados obtenidos de la experimentación del **Capítulo 6: Pruebas del prototipo**, la relevancia de los personajes se convierte en un factor fundamental para la identificación de historias secundarias, por lo que su valor es de un 34% mientras que el valor de la longitud de los capítulos, es decir, la cantidad de escenas de una historia y la cantidad de lugares visitados es del 33%. El cálculo del valor de importancia de una historia se realiza según la fórmula de la figura 30.

$$\frac{(PC * LC) + (PP * IP) + (PL * LV)}{3}$$

Figura 30: Fórmula empleada para calcular la relevancia de las historias y así poder clasificar las historias

- PC: Ponderación de la variable de capítulos
- LC: Longitud de capítulos
- PP: Ponderación de la variable de relevancia de personajes

- IP: Importancia de Personajes
- PL: Ponderación de la variable de cantidad de lugares visitados
- LV: Lugares visitados.

Previamente a la implementación de este sistema, se realiza una serie de pruebas reflejados en el **Capítulo 6: Pruebas del prototipo** para comprobar que el sistema identifica las historias de la misma manera en la que estaban diseñadas.

4.4.4 Sistema de peticiones con Unity

Una vez identificadas las historias, se ajustan las funciones encargadas de enviar la información a Unity cuando este la necesite. Igual que en la anterior versión, todas las historias se almacenan en una lista en la que la historia almacenada en la primera posición de la lista es la principal, la diferencia ahora es que se ha almacenado esa información según el nivel de relevancia calculado mediante el sistema de clasificación de historias.

En el caso de las historias secundarias, se plantea la posibilidad de que el juego tenga un gran número de historias secundarias y esto puede hacer que el sistema sea menos eficiente al tener que cargar mucha información en el caso de cargar todas las historias secundarias que se inician en un lugar. Por ello, cuando el sistema comprueba el número de historias secundarias que se inician en un lugar, este solo envía al juego un número aleatorio de historias entre los valores 1 y 3 para limitar esa carga.

Capítulo 5: Implementación

5.1 Tecnologías usadas en el desarrollo

Para la implementación del proyecto, se tuvo que determinar las tecnologías a utilizar. Se necesitaba un entorno de trabajo en el que fuera rápido construir y probar, para así avanzar lo antes posible a la fase de pruebas y cuestionarios, así como poder modificar el prototipo con los resultados obtenidos en las encuestas en mente.

Como ya se comentó en el **Capítulo 2.2.4 Herramientas de desarrollo**, para el desarrollo del prototipo se escogió Unity como entorno de trabajo. Es un motor de videojuegos que cumple con todas las características que se buscaban: Construir en él es rápido y más simple que otras opciones, lo que permitió al equipo crear un videojuego sencillo donde probar el motor. Además de ello, la experiencia previa de todos los miembros del equipo en su utilización ahorró un potencial proceso de aprendizaje que exigirían otras opciones.

El lenguaje de programación utilizado, tanto en el prototipo como en el motor, ha sido C#. El motivo de su uso ha sido por ser el lenguaje de programación de los guiones de Unity, y así no se necesitó programar en diferentes lenguajes. El motor podría haberse desarrollado en cualquier otro lenguaje de programación.

5.2 Implementación inicial del prototipo de estudio

El prototipo con el que se probarían las hipótesis fue el primero en desarrollarse. Un entorno bidimensional de perspectiva cenital, dando solo dos ejes de movimiento para simplificar el desarrollo. Mientras que Unity se encarga de gestionar la existencia inicial de los objetos del estado de juego (tales como el mapa y los personajes) y la funcionalidad de los componentes predefinidos, la funcionalidad específica ha sido implementada mediante varios componentes script (guiones). Cada sistema tiene al menos un gestor que se encarga de manejar el funcionamiento del resto de componentes.

- GameManager: Gestor del funcionamiento inicial del juego. Su función principal es la de tomar la ruta de datos del proyecto, donde se encuentran los archivos JSON, y llamar a la iniciación del motor (NarrativeEngine).
- PlayerMovement: Guión que implementa el movimiento del jugador.
 - float movespeed: velocidad constante de movimiento.
 - Vector2 movement: Dirección de movimiento actual.
 - Obtiene acceso a su Rigidbody en Start().
 - Movement(Context): Lee el valor de input de direcciones y lo convierte en una dirección de movimiento.
 - FixedUpdate(): Acción que se ejecuta en cada intervalo del sistema de físicas de Unity. Desplaza al jugador en la dirección indicada a una velocidad constante.
- Inventory: Clase que gestiona la posesión de objetos por parte del jugador.
 - List<Item> keyItemList: Lista de objetos clave.
 - Dictionary<Item, int> itemList: Diccionario que contiene objetos y la cantidad de ellos que se posee.
 - AddItem(Item item): Añade un objeto dado a su respectivo contenedor, o incrementando su cantidad en caso de objetos comunes.
 - RemoveItem(Item item): Reduce la cantidad del objeto dado en 1, o lo elimina de la misma en caso de ser clave o el único que queda en el inventario.
 - bool FindItem(Item item): Devuelve si el objeto (clave o consumible) se encuentra dentro de su respectiva lista.
 - UseItem(Item): Aplica el efecto del objeto llamando a Item.Effect(). En caso de ser consumible, remover uno con RemoveItem().
- enum ItemType:
 - Key: Representa objetos clave, objetos únicos que no se pueden consumir.
 - Consumable. Representa objetos consumibles, objetos con cantidad variable.
- Item: Componente que representa un objeto genérico.

- `string itemName`: Nombre del objeto.
- `ItemType type`: tipo del objeto.
- `Effect()`: Función genérica que aplicaría el efecto del objeto. En esta clase no hace nada, y cada clase hija redefine el método.

5.2.1 Sistema de interacciones

Sistema encargado de las interacciones del jugador con el entorno. Mediante la lectura de entrada y proximidad, el jugador puede activar puertas o hablar con personajes.

- **PlayerInteractionManager**: Gestor del sistema de interacciones. Registra el objeto interactuable más cercano y lo llama cuando recibe la entrada correspondiente.
 - `Interactable objectToInteract`: Objeto interactuable en contacto con el jugador.
 - `InputReceived(Context)`: Cuando hay un objeto interactuable y no se encuentra en mitad de una conversación, indica la existencia de entrada al objeto correspondiente.
- **Interactable**: Clase padre que denota a un objeto interactuable.
 - Tiene referencias al objeto *player* y al `PlayerInteractionManager`, los cuales busca y registra en `Start()`.
 - `Interact()`: Método virtual de interacción de la que heredan las clases hijas. En esta clase no hace nada.
 - `InputReceived()`: En caso de estar dentro de su rango, ejecuta su interacción y lo quita de `PlayerInteractionManager` para no provocar una interacción doble.
 - `OnCollisionEnter2D(Collision2D)`: Si el objeto interactuable colisiona con el jugador, se asigna a sí mismo como objeto a interactuar más cercano.
 - `OnCollisionExit2D(Collision2D)`: Se elimina a sí mismo de su referencia en `PlayerInterInteractionManager` una vez deja de colisionar con el jugador.
- **Door**: Componente hijo de `Interactable` que cumple la función de puerta.
 - Referencia a su puerta de destino.

- Interact(): Recoloca al jugador en la posición de la puerta de destino, añadiendo un offset para no aparecer encima de la misma.
- NPC: Componente hijo de Interactable que cumple la función de un personaje no jugador. Disponen de un sistema de movimiento propio.
 - Puede disponer de diálogos genéricos, conferidos por el sistema de diálogos.
 - Posee parámetros para regular dirección, distancia máxima y velocidad de movimiento, así como periodo de tiempo entre cada paso.
 - Interact(): Si posee diálogo genérico, solicita su ejecución a DialogManager.
 - Update(): Si el personaje no es visible, llama a InvisibleBehaviour(). En cualquier caso, llama a WanderingBehaviour().
 - FixedUpdate(): Llamado en cada frame del motor de físicas. Llama a Move().
 - DialogEnded(bool): Método virtual que indica comportamiento una vez acaban los diálogos. En esta clase no hace nada, pero será sobrescrita en clases hijas.
 - Teleport(Place): Indica al personaje que debe teletransportarse cuando se encuentre fuera de cámara.
 - InvisibleBehaviour(): Comportamiento del personaje cuando está fuera de cámara. En caso de necesitar teletransportarse mueve al personaje a una de las coordenadas disponibles en el lugar destino, y es transferido de una lista de personajes a la otra.
 - bool NeedsToTeleport(): Comprueba con el booleano del mismo nombre si necesita teletransportarse.
 - WanderingBehaviour(): Comprueba si es hora de moverse y si está quieto sin interactuar, y de ser así solicita una nueva dirección. En cualquier caso, de tocar moverse calcula una nueva cuenta atrás para el próximo movimiento.
 - NewWanderingDirection(): Calcula una nueva dirección de movimiento aleatoriamente, cualquiera de las cuatro direcciones cardinales.
 - Move(): De estar moviendo o tocar moverse, y no está interactuando, calcula la distancia de la coordenada origen, e invierte la dirección de movimiento en

caso de estar demasiado lejos. Inmediatamente después desplaza al personaje a en dirección dada.

- `OnCollisionEnter2D(Collision2D)`: Si puede moverse y se choca con el jugador, se prepara para interactuar con él y se queda quieto. En caso contrario invertirá su dirección de movimiento. En cualquier caso, seguirá con la implementación de la clase `Interactable`.
- `OnCollisionExit2D(Collision2D)`: Si termina de estar en contacto con el jugador, determina que ha terminado de interactuar con él y puede seguir moviéndose. Sigue con la implementación de la clase `Interactable`.

5.2.2 Sistema de diálogos

Sistema que se encarga del funcionamiento de todos los diálogos del juego. Utiliza funcionalidad ya implementada en Unity para la carga de texturas y textos, mientras que los guiones programados para este proyecto se encargan de activarlos y desactivarlos, así como alterar el texto para mostrar el diálogo que se desea cargar.

- `DialogManager`: Gestor de diálogos de cualquier escena del juego, así como diálogos genéricos.
 - `Narrative_Engine.Dialog dialog`: Diálogo actual.
 - `NPC character`: personaje que inicializa el diálogo.
 - `Text characterName`: Nombre del personaje hablando en el momento actual.
 - `Text displayedText`: Texto mostrándose actualmente en pantalla.
 - `Narrative_Engine.Node currentNode`: Opciones y texto actuales.
 - `Awake()`: Se asegura de que solo exista una instancia de `DialogManager`.
 - `IsOnDialog()`: Determina si hay un diálogo en proceso.
 - `OnClick(Context)`: Avanza un texto si había un diálogo en proceso, mostrando opciones a elegir en caso de necesitarlo. Si se acaba de escoger una opción, se carga el diálogo correspondiente a la opción escogida.
 - `OnInput(Context)`: Comprueba que se encuentra en una lista de opciones, y desplaza el cursor en la dirección de la tecla indicada. En caso de que se vaya a desplazar fuera de la lista de opciones, cargar nuevas opciones de darse el caso con `SelectorAtTop()` o `SelectorAtBottom()`.

- `SelectorAtTop()`: Carga el grupo anterior de opciones. En caso de ser el primer grupo, pasa al último grupo de opciones en su lugar, y coloca la flecha en la opción inferior.
- `SelectorAtBottom()`: Carga el grupo siguiente de opciones. En caso de ser el primer grupo, pasa al primer grupo de opciones en su lugar, y coloca la flecha en la opción superior.
- `StartDialog(Dialog, index, NPC)`: Si no se encuentra ya en mitad de un diálogo asigna toda la información del diálogo dado al gestor, así como el personaje e índice. Llama al primer diálogo con `NextDialog()`.
- `NextDialog(index)`: Carga diálogos con una funcionalidad determinada.
 - Si el índice es menor de -1, llama a `NPC.DialogEnded()`.
 - -2: El diálogo ha acabado, y la misión puede avanzar.
 - -3: El diálogo ha terminado sin aceptar el inicio o progreso de misión. Esto permite al diálogo de la escena empezar desde el principio la próxima vez.
 - En otro caso, solicita la carga del nodo de diálogo siguiente.
- `ShowOptions()`: Activa el sistema de selección de opciones e instancia todas las opciones necesarias llamando a `InstantiateOptions()`. Finalmente coloca y hace visible el selector.
- `LoadNextOptionPack()`: Obtiene el conjunto de opciones que deben cargarse y sustituye el texto presente por el correspondiente a las opciones.
- `InstantiateOptions()`: Obtiene la cantidad total de opciones del diálogo actual y las divide en grupos de tres (paquetes). Finalmente solicita la carga del primer paquete con `LoadNextOptionPack()`.
- `LoadDialogues(Narrative_Engine.StoryScene)`: Solicita a `NarrativeEngine` los diálogos de la escena del motor dada como parámetro, llamando al `LoadDialogues()` del motor.
- `LoadGenericDialogsByPlace(place)`: Solicita al motor los diálogos genéricos que ocurren en el lugar dado al llamar a `NarrativeEngine.loadGenericDialogs()`, y se lo asigna a los personajes correspondientes. En caso de no tener un componente que herede de `NPC`, crea uno de clase `NPC` y se lo asigna.

5.2.3 Sistema de misiones

El sistema encargado de gestionar la historia de juego, dividida en misiones. Las misiones son cargadas, asignadas a personajes, inicializadas y modificadas por cada interacción del jugador.

- QuestManager: Gestor de misiones del juego. Se encarga de crear las misiones tras su carga, de asignarlas a los personajes correspondientes, y avanzar su estado por cada llamada desde el resto del prototipo.
 - `StartQuest(Quest)`: Inicia la primera escena de la misión dada. Si se necesita entregar un objeto, se añade al inventario.
 - `DoScene(Quest)`: Inicia la escena actual de la misión dada. En caso de necesitar un objeto, se solicitará al jugador. Si se tiene en el inventario, la misión seguirá. En caso contrario se abortará. La misión dará un objeto al jugador si la misma lo tiene indicado.
 - `LoadQuest(Narrative_Engine.Quest)`: Lee un capítulo/misión del motor y lo convierte en una misión del juego.
 - Crea un objeto de la clase `Quest` de Unity, y recorre todas las escenas. En caso de ser la primera escena, la marca como primera escena en la misión. En el resto de casos, enlaza secuencialmente la escena previa con la actual. Asigna todos los datos pertinentes de una escena del motor, a su equivalente de Unity. y solicita a `DialogManager` la carga de sus diálogos.
 - Por cada diálogo de las escenas del motor:
 - En caso de ser un pensamiento (el dueño del diálogo es el jugador), asignarlo al lugar.
 - En otro caso, asignarlo al personaje del mismo nombre.
 - Convertir al personaje en un encargado de misiones con `AssignCharacter()`. Si el personaje no se encuentra en su lugar correspondiente, solicitar su teletransporte con `NPC.Teleport()`.
 - En caso de ser un pensamiento, generar un gatillo con `Place.CreateTrigger()`.

- NPC `AssignCharacter(StoryScene, GameObject character, boolean, sceneNumber, dialogIndex)`: Asigna los datos de la misión al personaje, modificando o creando un componente hijo de NPC. Si es una escena de inicio crea o modifica `QuestStarterNPC`, o `QuestFinisherNPC` en cualquier otro caso.
- **Quest**: Clase que representa una misión, y un capítulo de historia.
 - `List<StoryScene> storyScenes`: Lista de escenas de la misión.
 - `int sceneCount`: posición de la escena actual en la lista.
 - `BeginScene(StoryScene scene)`: Solicita la escena empezar con sus diálogos con `StoryScene.ConsumeDialog()`.
 - `StoryScene CurrentStoryScene()`: Devuelve la escena actual dentro de la misión.
 - `ProgressQuest()`: Si la escena puede avanzar, indica la escena como terminada y avanza a la siguiente. En caso de haber sido la última escena la considera como finalizada, y solicita al motor la siguiente misión de la misma historia, solicitando a `QuestManager` que la cargue en caso de existir.
 - `bool CanProgressQuest()`: Determina si una misión puede avanzar. La última escena no debe haber terminado, el jugador ha salido del lugar de la escena actual, y todos los diálogos de la escena han sido consumidos.
- **StoryScene**: Clase que representa una escena individual dentro de una misión.
 - `Place place`: lugar en el que ocurre la escena.
 - `Quest quest`: Misión a la que pertenece la escena.
 - `bool dialogConsumed`: Indica si el diálogo de la escena ha sido consumido.
 - `ConsumeDialog()`: Asigna verdadero a `dialogConsumed`, indicando que el diálogo de la escena ha sido consumido.
 - `bool AreDialogsConsumed()`: Devuelve `dialogConsumed`.
- **QuestStarterNPC**: Componente hijo de NPC utilizado para el sistema de misiones.
 - `Interact()`: Si la misión está sin iniciar ni completada, y si el diálogo no está consumido, solicita el capítulo equivalente al motor y pide a

DialogManager que inicie el diálogo correspondiente a la primera escena. En caso de no tener que empezar la misión, llama al Interact() de NPC para su diálogo genérico.

- DialogEnded(boolean): Asigna el estado del booleano (verdadero si el diálogo ha acabado) al inicio de la misión. En caso de haber acabado el diálogo, da la misión por activa y solicita a QuestManager que inicie la misión. Si es la segunda vez que se intenta terminar un diálogo, transmite la información a QuestFinisherNPC en caso de existir.
 - Update(): Ejecuta Update() de NPC, y llama a Quest.ProgressQuest().
- QuestFinisherNPC: Componente hijo de NPC utilizado para el sistema de misiones. Es utilizado para cualquier escena que no sea la inicial de una misión.
 - Interact(): Si la misión está inicializada, sin terminar, toca la escena correspondiente al personaje, y el diálogo no está consumido, solicita el capítulo equivalente al motor y pide a DialogManager que inicie el diálogo correspondiente a la escena actual. En caso de no tener que avanzar la misión, llama al Interact() de NPC para su diálogo genérico.
 - DialogEnded(boolean): En caso de que el diálogo de la escena no haya sido consumido aún le asigna el estado dado por el booleano. Verdadero da el diálogo por concluido.
 - Update(): Ejecuta Update() de NPC, y llama a Quest.ProgressQuest().
 - bool NeedsToTeleport(): Determina si el personaje necesita teletransportarse. Además del resultado de NPC.NeedsToTeleport(), tiene que tocar la escena asignada al personaje.
- Place: Componente que representa un lugar en el mapa.
 - List<GameObject> characters: lista de personajes en el lugar.
 - Vector2 upLeftCoordinates: Vector que determina el límite superior e izquierdo del lugar en el mapa.
 - Vector2 downRightCoordinates: Vector que determina el límite inferior y derecho del lugar en el mapa.

- `bool loaded`: Determina si las escenas de las historias iniciales del área han sido cargadas.
 - `QuestTrigger questTrigger`: Referencia al gatillo de misión en caso de poseerlo. Por defecto es `null`.
 - `Update()`: Si el jugador está dentro del área del lugar, carga las misiones y los diálogos genéricos en caso de no haber cargado nada. Además, si hay un gatillo en el lugar le solicita su ejecución al gatillo de misión, y en caso de que el gatillo se desactive, lo destruye.
 - `bool PlayerIsInPlace()`: Comprueba si el jugador está dentro del área del lugar, mediante las coordenadas dadas.
 - `LoadQuests()`: Solicita a `NarrativeEngine` una lista de todos los capítulos/misiones que empiezan en ese lugar, y solicita a `QuestManager` cargar cada uno de ellos.
 - `CreateTrigger(NPC npc)`: Llamado por `LoadQuest()` de `QuestManager` cuando una escena requiere de un pensamiento. Crea un gatillo de misión y se lo asigna al lugar. Un personaje (que suele ser `QuestStarterNPC` o `QuestFinisherNPC`) es asignado al gatillo de `QuestTrigger`.
 - `LoadGenericDialogs()`: Solicita a `DialogManager` la carga de diálogos genéricos.
- **QuestTrigger**: Componente encargado de activar interacciones fuera del sistema usual de interacción mediante un gatillo.
 - `NPC trigger`: Referencia al personaje que va a interactuar con el jugador como gatillo.
 - `ActivateTrigger()`: Ejecuta el `Interact()` del gatillo. Una vez realizado, desactiva el gatillo de misión.

5.3 Implementación inicial del motor

El motor está desarrollado en el lenguaje de programación C#, específicamente en la plataforma de destino .NET Core 3.1. Como ya se mencionó, se utilizó este lenguaje para tener una consistencia con Unity durante el proceso de desarrollo. Además, también se hizo

por motivos de compatibilidad, pues es menos laborioso utilizar librerías de C# que otros lenguajes, al ser este el lenguaje de los scripts de Unity. NET Core 3.1 fue escogida porque Unity está programado para ser compatible con dicha plataforma, pues versiones más recientes como .NET Core 5 son completamente incompatibles.

El motor es compilado como una librería dinámica, y añadida junto a otras librerías externas a la carpeta de Assets del proyecto en Unity. Esto permite a los scripts acceder al motor como si fuera una librería más con solo incluirla. Posteriormente, se tendrá que dar la ruta de localización de la carpeta JSON que incluye todos los datos en sus archivos, y podrá proceder a leerlos y cargarlos internamente.

Se procederá a explicar su implementación para la primera versión del prototipo, que utiliza el espacio de nombres `Narrative_Engine`.

5.3.1 Sistema de lectura de ficheros

Originalmente para la lectura de ficheros se utilizaba la implementación incluida en .NET Core 5 (`System.Text.JSON`). Al ser .NET Core 5 incompatible con Unity, esta implementación de lectura de ficheros fue descartada y sustituida por la siguiente implementación:

- **SimpleJSON**: Librería externa para la lectura y escritura de ficheros en el formato JSON. La funcionalidad utilizada de esta librería es la siguiente:
 - `JObject`: Clase interna que contiene toda la información de una estructura JSON. Tiene varias funciones para extraer información de la misma en el tipo de dato deseado, como arrays, diccionarios, strings, booleanos y números.
 - `JSONDecoder.Decode(stream)`: Dado un flujo de texto en formato JSON, lo convierte en una estructura de clase `JObject`.
- **FileManager**: Clase interna encargada de leer los ficheros en formato JSON y transformar sus datos en los tipos internos del motor.
 - Posee como miembros strings de las rutas de todo lo que tiene que buscar, tanto archivos como sub-carpetas.
 - `List<string> characters`: Lista que contiene los nombres de todos los personajes involucrados en las historias.

- `List<string> items`: Lista que contiene los nombres de todos los objetos involucrados en las historias.
- `FileManager()`: Constructor que recibe la ruta general de los archivos JSON, y los nombres de todos los archivos y sub-carpetas. Con ello calcula las rutas finales de todos los archivos principales y sub-carpetas.
- `ReadFiles()`: Como funcionamiento general, Lee archivos de texto mediante el sistema de entrada y salida de C#, y los convierte en flujo de texto, guardados en strings. Llama a `JSONDecoder` para decodificar los strings y convertirlos en `JObject`, y extrae información del tipo adecuado cuando se necesita de ellos.

Los personajes y objetos los guarda en sus listas correspondientes. Para lugares, crea listas con nombres de lugares adyacentes y diálogos genéricos que posee. Con ellos crea el lugar (`Place`) y añade el mismo a la lista de lugares de `PlaceManager`. Posteriormente pide a `PlaceManager` que complete la lista de adyacentes.

Las historias tienen un proceso más laborioso: Hay 3 archivos, uno para historias, otro para capítulos/misiones, y otro para escenas. Cada uno tiene una lista con nombres de los componentes que poseen (una historia tiene una lista de capítulos, una misión tiene una lista de escenas, y una escena tiene una lista de diálogos). Crea objetos de sus respectivas clases y los añade a sus contenedores correspondientes de `StoryManager`.

- `Dialog ReadDialogFile(filePath)`: Dada la ruta de un archivo JSON con un diálogo, lo crea y lo devuelve. Del archivo lee la lista de nodos, y por cada nodo carga sus opciones (que pueden estar vacíos), así como el índice de diálogo de las respuestas. Con todo eso crea las opciones, nodos y diálogo, y devuelve el diálogo nuevo..
- `string[] LocateDialogFiles()`: Usando la ruta a la subcarpetas con los archivos de diálogo, solicita al sistema de entrada y salida que le de una lista de todos los archivos de extensión `.json`. Devuelve un array con todas las rutas individuales de cada uno de los archivos.

5.3.2 Sistema del motor

El sistema principal, el cual contiene todo lo importante del proyecto. La mayoría de clases son internas, por lo que alguien que quiera acceso al motor solo podría acceder mediante NarrativeEngine. Otras clases como Dialog son usadas externamente como una implementación común, pero no pueden acceder a funcionalidades internas y solo a los datos.

- **NarrativeEngine**: Clase pública que hace de frente exterior del motor. Esta es la clase que es utilizada para la comunicación entre cualquier juego y el motor. Tiene referencias al DialogManager y a FileManager.
 - `Init(generalPath)`: Crea un objeto de clase FileManager mediante la ruta dada por generalPath. También indica que se buscan "Story.json" para las historias, "Chapters.json" para los capítulos/misiones, "Scenes.json" para las escenas, "Dialogs" para la carpeta de diálogos, y "Characters.json", "Items.json" y "Place.json" para los personajes, objetos y lugares respectivamente. Solicita la lectura de ficheros a FileManager, y en ensamblado de las mismas a StoryManager para que haya una referencia directa Story > Quest > StoryScene. Por último crea el DialogManager, dándole una referencia al FileManager.
 - `List<Quest> GetChaptersByPlace(place)`: Solicita al PlaceManager la lista de misiones iniciales de un lugar dado.
 - `Quest getChapterById(chapterId)`: Dado el nombre de la misión, solicitarla al StoryManager y devolverla.
 - `Quest GetNextChapterById(chapterId)`: Dado el nombre de una misión, solicitar al StoryManager la siguiente y la devuelve.
 - `LoadDialogues(StoryScene)`: Solicita al DialogManager la carga de todos los diálogos de la escena dada.
 - `List<Dialog> LoadGenericDialogs(place)`: dado el nombre de un lugar, solicitar al DialogManager la carga de los diálogos genéricos en dicho lugar según PlaceManager, y devuelve la lista de diálogos.
 - `List<Story> GetStories()`: Obtiene la lista de historias en StoryManager y la devuelve.

- Dictionary<string, Quest> GetQuests(): Obtiene el diccionario de capítulos/misiones en StoryManager, y lo devuelve.
- Dictionary<string, StoryScene> GetStoryScenes(): Obtiene el diccionario de escenas en StoryManager, y lo devuelve.
- PlaceManager: Gestor interno de los lugares.
 - Dictionary<string, Place> places: Diccionario de lugares, archivados por su nombre.
 - Dictionary<string List<Quest>> questsInPlace: Diccionario de listas de misiones, archivadas por el nombre del lugar al que pertenecen.
 - CompleteQuestsInPlace(): Solicita a NarrativeEngine las historias, y crea listas de misiones para cada uno de los lugares, archivándolas por el lugar en el que empiezan sus historias.
 - CompletePlaces(): Solicita a cada lugar que se complete su lista de lugares adyacentes.
 - GetGenericDialogsInPlace(place): Dado el nombre de un lugar, devolver su lista de diálogos genéricos.
 - List<Quest> GetQuestsInPlace(place): Dado el nombre de un lugar, devuelve las misiones que empiezan allí.
- StoryManager: Gestor interno de las historias.
 - List<Story> stories: Lista de todas las historias.
 - Dictionary<string, Quest> chapters: Diccionario que guarda capítulos/misiones, archivados por su id.
 - Dictionary<string, StoryScene> storyScenes: Diccionario de escenas, archivadas por su id.
 - AssembleStory(): Toma los nombres de cada capítulo en una historia, y añade una referencia a su misión correspondiente dentro de la historia, para todas las historias. Durante el proceso hace lo mismo con las escenas, añadiendo a las misiones referencias a sus escenas.
 - Quest GetChapterById(chapterId): Dado un nombre de capítulo, devuelve su misión correspondiente.
 - Quest GetNextChapterById(chapterId): Dado un nombre de capítulo, devuelve la misión siguiente correspondiente.

- DialogManager: Gestor interno de los diálogos. Se encarga de cargar y guardar datos de cada uno de los diálogos en el momento que son necesarios. Tiene referencia al FileManager.
 - DialogFileData: Estructura interna que representa los datos de los archivos de diálogo.
 - string path: Ruta del archivo de diálogo.
 - bool consumed: Indica si el diálogo ha sido consumido o no.
 - Dictionary<string, DialogFileData> filesData: Diccionario que guarda la información sobre los archivos de los diálogos.
 - DialogManager(FileManager): Constructor. Solicita a FileManager una lista con las rutas de los archivos de diálogo, y crea una entrada de datos en el diccionario por cada ruta.
 - Dialog GetDialog(filename): Dado un nombre de archivo, busca su ruta en el diccionario y solicita a FileManager la carga de su diálogo correspondiente. Posteriormente devuelve el diálogo.

- Place: Clase interna que representa un lugar. Posee listas de lugares adyacentes y de diálogos genéricos de los personajes que se encuentran en ese lugar.
 - Place(name, adjacentPlacesNames, genericDialogs): Constructor. Solo asigna los parámetros dados, y el resto de miembros se obtienen con otros métodos.
 - SearchAdjacent(places): Dado el diccionario con todos los lugares archivados por su nombre, busca el lugar al que está conectado y lo añade a su lista de lugares adyacentes.

- enum StoryType:
 - Main: Representa la historia clasificada como principal. Solo habrá una.
 - Secondary: Representa las historias clasificadas como secundarias.

- Story: Clase pública que representa una historia completa. Tiene el tipo de historia al que pertenece, y una lista con sus capítulos/misiones.
 - AddQuest(Quest): Añade la misión dada a la lista.

- Quest: Clase pública que representa los capítulos/misiones que componen una historia. Además de su id, tiene una lista de escenas y el índice de la escena actual, así como el id de la misión que la sigue.
 - `AddScene(StoryScene scene)`: Añade la escena dada a la lista.
- StoryScene: Clase pública que representa las escenas que componen una misión. Además de su id, tiene una lista de diálogos, y objetos involucrados en la misma, tanto para dar como a recibir.
 - `StoryScene(...)`: Constructor interno. Asigna todos los valores, y solo puede ser usado por el motor..
- Dialog: Clase pública que representa un diálogo que ocurre en una escena particular.
 - `string init`: Nombre del personaje que inicia el diálogo.
 - `List<Node> nodes`: Lista de nodos, que contienen información sobre cada parte del diálogo.
- Node: Clase pública que representa unidades de información sobre el diálogo al que pertenecen. Tiene referencias al personaje que habla, índice del nodo siguiente en la conversación, el texto a mostrar, y la lista de opciones de diálogo en caso de haberlas.
- Option: Clase pública que representa una de las opciones a tomar durante una conversación.
 - `int nodePtr`: Índice a su propio nodo.
 - `text`: Texto que se mostrará en pantalla.

5.4 Cambios en la implementación tras el experimento

Tras la experimentación de la que se hablará en el **Capítulo 6: Pruebas del prototipo**, se tuvo que cambiar algunas partes del código, tanto del motor como del prototipo, para reflejar la nueva hipótesis y a su implementación en el propio sistema.

5.4.1 Cambios en el motor

En el motor, los cambios principales se centran en implementar la hipótesis, así como leer correctamente los archivos JSON modificados para tal fin. En términos generales, se obtienen y calculan los valores para calcular la media ponderada de la importancia de una historia, y en consiguiente ordenar todas las historias por orden de importancia. Así mismo, se ha modificado el código para aplicar el cambio a la hora de carga de historias, ya que ahora se solicitan una cantidad reducida de historias nuevas cada vez que se visita un lugar, en vez de cargarlas todas a la vez.

- QuickGraph: Librería externa utilizada para programar la hipótesis de lugares visitados. Se crea un grafo cuyos nodos son los lugares, y sus caminos las aristas. Se utiliza el algoritmo de Floyd-Warshall para calcular cualquier camino para ir de un nodo a otro de la manera más rápida posible.
- FileManager:
 - Eliminación de `characters`, pues se ha expandido la implementación de los personajes.
 - Cambios en `ReadFiles()`: Ahora la lectura de personajes es más compleja, al ser ahora una clase completa y no solo un nombre. Va creando personajes, y añadiendo a un contador la importancia de cada uno. Todos son dados a `CharacterManager`. En la lectura de Lugares, se crea una estructura de grafo con `QuickGraph`, y se añade al `PlaceManager` junto a todos los lugares. Por último, se ejecuta el algoritmo de Floyd para calcular todos los caminos más cortos.

En la lectura de historias, capítulos y escenas, el archivo de historias es el último en ser leído. Durante su lectura, se buscan los capítulos correspondientes a la historia, y se calcula a partir de ellos el total de escenas que componen la historia. Así mismo, se calcula con la lista de personajes involucrados, el total de importancia de todos los personajes. Se crea la historia utilizando todos estos datos, y se guarda en un set de historias, ordenado por orden de importancia final de cada historia (las más importantes se encuentran más cerca del principio del set). Dicho set se encuentra en `StoryManager`.

- NarrativeEngine:
 - `List<Story> GetStoriesByPlace(place)`: Dado el nombre de un lugar, pide a `PlaceManager` el set ordenado de historias que empiezan en él. Selecciona una cantidad aleatoria de 1 a 3 historias más importantes, y las devuelve en una lista.
 - `Quest GetMainQuest()`: Adquiere la historia más importante (la cual sería la principal, la marca como consumida, y devuelve su primera misión.
 - Cambios en `List<Quest> GetChaptersByPlace(string place)`: Ahora solicita un set de historias que ocurren en ese lugar, y devuelve una lista con todos sus primeros capítulos.

- PlaceManager:
 - `Dictionary<string, List<Story>> storiesStartingInPlace`: Diccionario que contiene listas historias, archivadas por el nombre del lugar en donde empiezan.
 - `AdjacencyGraph<string, Edge<string>> graph`: Grafo dirigido que representa el mapa del juego, donde cada nodo es un lugar y cada arista las conexiones entre los mismos. También se tiene un objeto destinado a calcular mediante el algoritmo de Floyd y Warshall, y guardar, todos los caminos más cortos.
 - `HashSet<string> PathBetweenPlaces(source, target)`: Dados un lugar origen y su destino, se crea y se devuelve un set con todos los lugares que forman el camino más corto entre ambos.
 - `List<Story> GetStoriesInPlace(place)`: Devuelve una lista de todas las historias que empiezan en el lugar dado.
 - Cambios en `CompleteQuestsInPlace()`: Ahora marca todas las historias como secundarias excepto la más importante, que la marca como principal, y añade la historia a la lista de historias, en su lugar correspondiente.

- StoryManager: La lista de historias pasa a ser un set ordenado. El orden de la misma está determinado por la importancia final de todas las historias.

- CharacterManager: Nueva clase gestor que se encarga de gestionar el almacenamiento de los personajes.

- `Dictionary<string, Character> characters`: Diccionario de personajes, archivados por su nombre.
 - `uint totalImportance`: Importancia total de todos los personajes. Es asignado durante la lectura de ficheros.
- **Story**: La clase empieza a gestionar su nivel de importancia, siguiendo los valores de la hipótesis. Tiene valores de importancia total de personajes, cantidad de lugares visitados, y cantidad de escenas totales, así como un valor final de importancia que se calcula durante la construcción de la historia. Así mismo, implementa una interfaz para la comparación de historias mediante su nivel de importancia final.
 - `CalculateImportance()`: Calcula la importancia final mediante una media ponderada, utilizando los valores de la hipótesis comparados con la cantidad total de todo el juego (por ejemplo, la proporción de lugares visitados respecto a todos los que hay).
- **Character**: Nueva clase interna que encapsula la nueva información de los personajes según la hipótesis.
 - `string characterName`: Nombre del personaje.
 - `uint relevance`: Importancia del personaje, determinada por el autor de la historia.

5.4.2 Cambios en el prototipo

Para el prototipo, los cambios fueron menores. Muchas clases han sido limpiadas para hacer más legible el código, mientras que unas pocas han sido adaptadas a utilizar la nueva hipótesis:

- **GameManager**: Solicita al motor la primera misión de la historia principal, y solicita su carga a `QuestManager`, durante el proceso de inicio.
- **Place**: La carga se hace más compleja. Cuando el jugador llega a una zona por primera vez, solicita la carga de los diálogos genéricos. Sin embargo, ahora solicita en todas las visitas cargar nuevas historias (de 1 a 3 aleatoriamente), hasta que dicho lugar se queda sin historias que cargar.

Capítulo 6: Pruebas del prototipo

Para poder probar que la hipótesis de partida era correcta se lanzó una primera prueba piloto con una versión del juego en la que las historias estaban clasificadas y diseñadas por su tipo, para probar cada una de las variables que componen la hipótesis de partida. De esta manera se podría saber gracias a los diversos sujetos que realizaron la prueba qué historia reconocían como principal y cuáles como secundarias en relación a la hipótesis.

De forma que los resultados del experimento fuesen lo más objetivos posibles, se organizó el juego de tal manera que no se pudiera saber de primera mano cuál era la historia principal y cuáles las secundarias, disponiendo el jugador de todas ellas al comienzo del juego.

Para iniciar cada una de las historias, el jugador debería hablar con todos los personajes no jugables disponibles en la plaza principal del juego, iniciando de esta manera las historias y las misiones que contienen. Una vez finalizado el juego o en el momento en el que el jugador crea que ha completado todas las historias disponibles, completa el cuestionario que centra sus preguntas en la hipótesis de partida.

Gracias a ese cuestionario se obtendría la información necesaria para saber por un lado si la hipótesis es correcta y, de no ser así, qué clase de ajustes habría que hacer para que a la hora de implementar el motor con la hipótesis la clasificación fuese lo más precisa posible. Para ello, el cuestionario se organizó en diferentes secciones que preguntaban acerca de detalles del juego y las diferentes historias.

En primer lugar, se preguntó el número de historias identificadas durante la sesión de juego. Para comprobar que la identificación ha sido correcta, se pide al jugador una breve descripción de cada historia para confirmar que efectivamente las ha jugado, después otra pregunta sobre los personajes que interfieren y la relación y coherencia entre las historias que han sido reconocidas en la partida.

La siguiente sección trata de analizar cada historia siguiendo las diferentes variables propuestas en la hipótesis de partida, preguntando a la persona encuestada sobre la longitud de la historia, la cantidad de personajes que interfieren en ella y el número de

lugares que se han visitado en cada una de las historias. Además de esto, se cuestionó a los sujetos sobre si las historias jugadas podrían ser principales pidiendo las razones por las que creían que una historia podría o no ser principal. De esta manera, se podrían obtener más datos y variables que no se habían tenido en cuenta para plantear la hipótesis de partida y complementarla de manera que la clasificación obtenida por el algoritmo que implementa la hipótesis fuera más precisa.

6.1 Resultados

Una semana después del lanzamiento del prototipo y el formulario, se procedió al análisis de los datos obtenidos en las encuestas realizadas a un total de 10 personas.

Realizando un análisis de forma general a los resultados, se puede observar que de las nueve personas que jugaron la historia 1, la que se diseñó para ser la historia principal, el noventa por ciento de ellas coinciden en que se trata de la narrativa principal del juego. A continuación, se procede a mostrar y analizar los datos obtenidos para cada una de las historias individualmente.

6.1.1 Historia 1

La Historia 1, que es la historia en la que se realiza una misión importante para el alcalde de Fyrst, ha sido jugada por el noventa por ciento de los jugadores (figura 31). Por tanto, de las diez personas que jugaron un total de nueve han jugado esta historia.

¿Has jugado esta historia?

10 respuestas

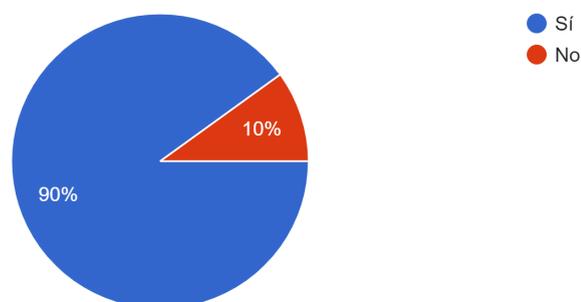


Figura 31: Porcentaje de jugadores que han jugado la historia número uno.

El 90% ha jugado frente a un 10% que no.

Al tener un alto porcentaje de jugadores, se puede decir que la historia se ha podido localizar con éxito y ha mantenido a la mayor parte de los jugadores interesados en la misma, además de que se ha podido seguir con éxito hasta el final.

En relación a la relevancia narrativa de la misma, de los 9 jugadores que han encontrado esta historia 8 consideran que es la historia principal (figuras 32 y 33). Puesto que la historia ha sido jugada por el 90% de los jugadores que han realizado la prueba, se puede decir que se ha interpretado esta historia tal y como estaba diseñada, es decir, para ser la principal del juego.

¿Es la historia principal?

9 respuestas

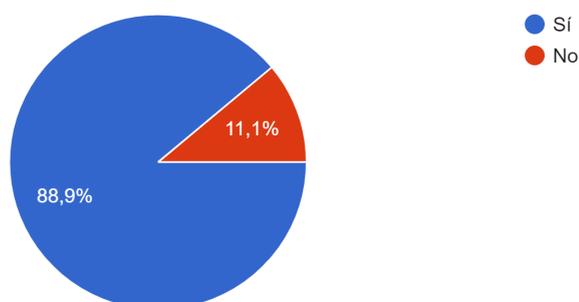


Figura 32: Porcentaje de jugadores que piensan que la historia número uno es la historia principal. El 88,9% piensa que sí frente a un 11,1% que piensa lo contrario.

¿Crees que esta historia podría ser la principal?

9 respuestas

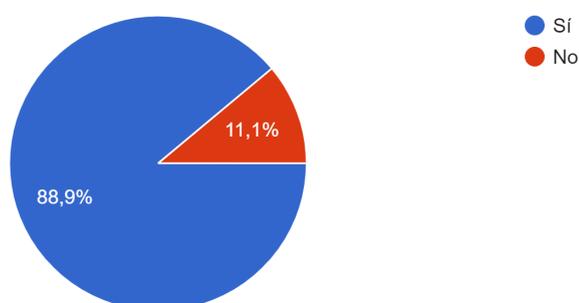


Figura 33: Porcentaje de jugadores que piensan que la historia número uno podría ser la historia principal. El 88,9% piensa que sí frente a un 11,1% que piensa lo contrario.

Junto a la pregunta a la que se corresponde el gráfico anterior, se preguntó también a los jugadores las razones por las que habían respondido una u otra opción en la anterior pregunta. El argumento que más se utilizó fue el de que esta historia poseía la cualidades necesarias para ser la principal. De este dato, luego se especificó más y se pudo extraer que las razones por las que se consideraba así era por la importancia narrativa de la historia, los personajes involucrados y los objetos que mueven toda la historia. Por ejemplo, al menos 5 personas mencionaron que el alcalde parecía un personaje muy importante y que por eso pensaban que la narrativa iba a ser la que más peso iba a tener.

Además de todo esto, 7 personas coincidían en que ésta era la principal porque era la primera que se encontraba nada más salir de la casa en la que se comienza el juego, por lo que la situación de inicio de la historia también fue relevante para reconocer el orden de importancia de las diferentes narrativas.

En cuanto a la duración de esta historia, se puede observar en la figura 34 que se ha considerado de duración media, teniendo todos los datos que se han obtenido en esta encuesta de manera que formen aproximadamente una distribución normal, contando con un máximo en el valor 3 de duración.

¿Te ha parecido muy larga?

9 respuestas

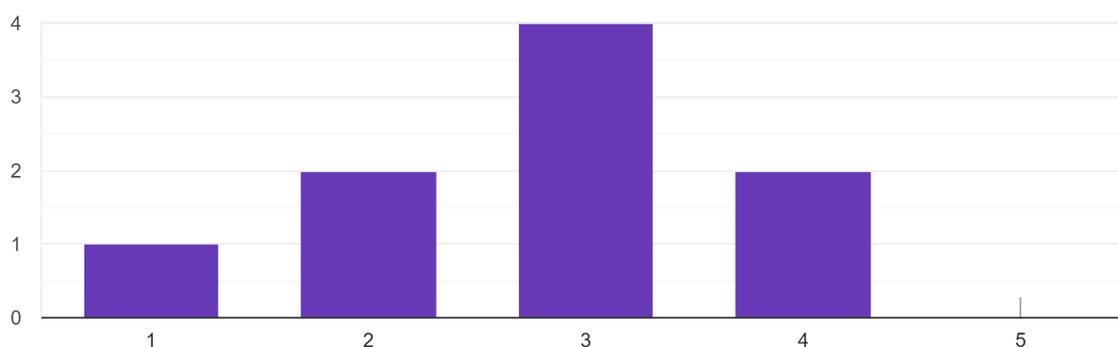


Figura 34: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

En cuanto a la cantidad de personajes, también cuenta con un máximo en el punto 3 con un total de 5 personas seleccionando la opción, como se puede ver en la figura 35. Por

ello, se puede decir que la cantidad de personajes es la necesaria para el correcto desarrollo de la narrativa.

¿Hay pocos o demasiados personajes involucrados?

9 respuestas

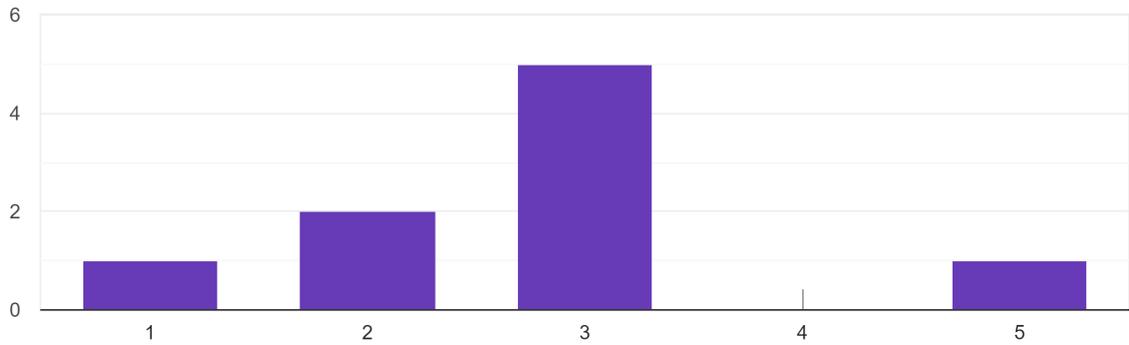


Figura 35: Porcentaje de jugadores que valoran la cantidad de personajes de la historia, siendo el valor 1 pocos personajes y 5 muchos personajes.

6.1.2 Historia 2

En esta historia el objetivo es encontrar a la nieta de un hombre que cuenta cómo se ha ido a buscar una espada legendaria con tal de convertirse en heroína. El porcentaje de jugadores que ha jugado esta historia es de un 90%, es decir, 9 de 10 personas han encontrado esta narrativa (figura 36).

¿Has jugado esta historia?

10 respuestas

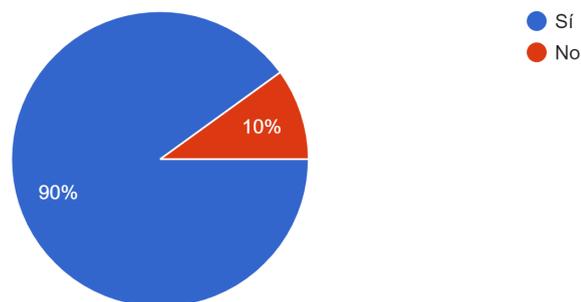


Figura 36: Porcentaje de jugadores que han jugado la historia número dos.

El 90% ha jugado frente a un 10% que no.

Esta historia también ha sido jugada por un gran número de personas, por lo que la historia es localizable incluso estando más separada de la zona inicial y la mayor parte de los jugadores han podido seguir sin problemas esta narrativa.

Los resultados de la figura 37 demuestran que ninguno de los jugadores que han encontrado esta historia ha considerado la misma como la narrativa principal del videojuego. Sin embargo con esta información no se puede saber si podría llegar a serlo o no y por qué. Por tanto, es necesario analizar los datos de la siguiente cuestión.

¿Es la historia principal?

9 respuestas

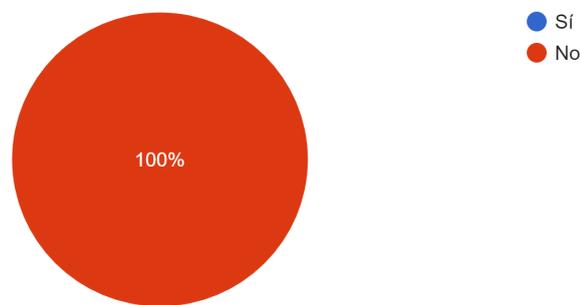


Figura 37: Porcentaje de jugadores que piensan que la historia número dos es la historia principal, siendo no para el 100% de los jugadores de la historia.

A la hora de plantearse la historia como posible principal, solo el 22,2%, es decir 2 personas, consideran que esta podría haber sido la principal (figura 38). De las personas que sí la han considerado como posible principal, mencionan que la consideran la más desarrollada de todas las historias. Sin embargo, el resto de personas argumentan que es demasiado corta como para llegar a ser la historia principal del juego.

¿Crees que esta historia podría ser la principal?

9 respuestas

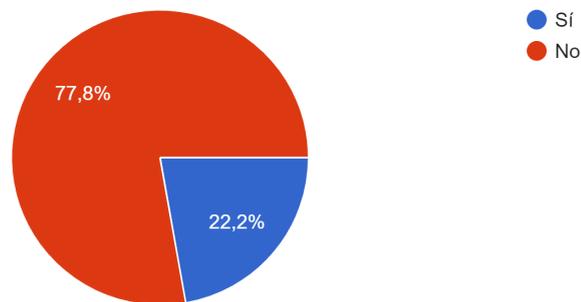


Figura 38: Porcentaje de jugadores que piensan que la historia número dos podría ser la historia principal. El 22,2% piensa que sí frente a un 77,8% que piensa lo contrario.

Además también se ha mencionado bastante que esta historia tiene un final bastante cerrado y que impediría su continuación, por lo que tampoco ha sido considerada como posible historia principal por ese otro motivo.

Al estudiar la duración se ha obtenido que, pese a ser igual en duración que la historia principal, es demasiado corta, teniendo el valor máximo en 2 con 4 personas votando ese nivel de duración (figura 39). Por tanto, se puede extraer de este dato que lo que afecta a la duración de la historia es otro factor.

¿Te ha parecido muy larga?

8 respuestas

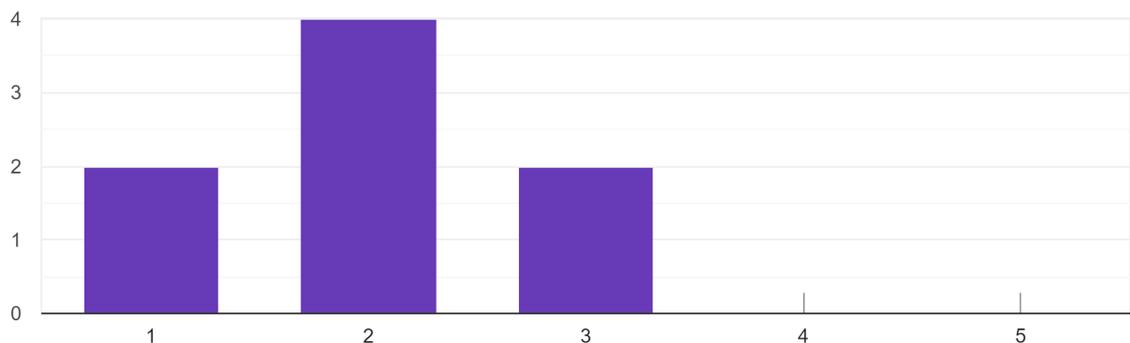


Figura 39: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

Observando las respuestas de los jugadores, se puede deducir que la duración de las historias se ve afectada por la calidad de esta y de sus personajes, coincidiendo con las observaciones obtenidas en la historia uno.

En cuanto a la cantidad de personajes, se sigue manteniendo una percepción media de cantidad de los mismos, teniendo a 6 personas que han votado una cantidad 3 de personajes y la cantidad de los mismos tampoco se ha mencionado mucho en las respuestas (figura 40).

¿Hay pocos o demasiados personajes involucrados?

9 respuestas

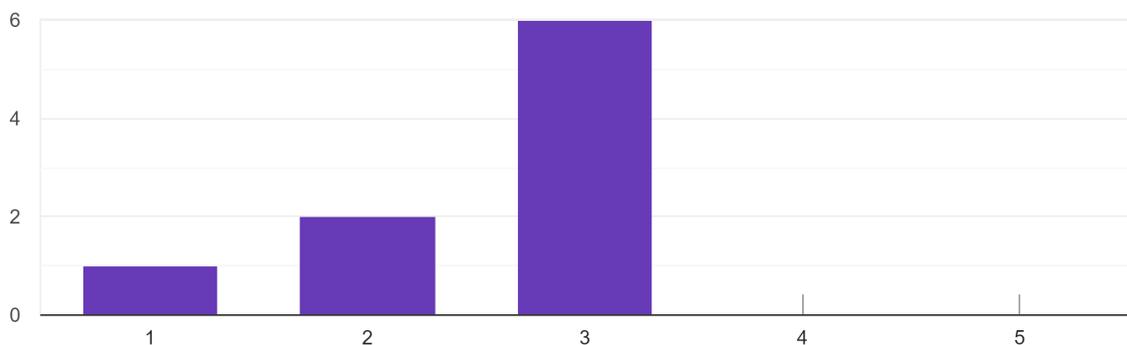


Figura 40: Porcentaje de jugadores que valoran la cantidad de personajes de la historia, siendo el valor 1 pocos personajes y 5 muchos personajes.

6.1.3 Historia 3

Esta historia transcurre en su gran mayoría en la Taberna de Dara, donde el jugador tendrá que relacionarse con diversos personajes para intentar reclutarlos en una misión que nos da una mujer en Fyrst. Está pensada para analizar el impacto del número de personajes en una narrativa a la hora de determinar si esta es principal o secundaria.

En este caso las cifras son un poco distintas a las dos historias anteriores. Sólo un 70% de los jugadores han interactuado con esta historia, como se puede ver en la figura 41. Esto quiere decir que 7 personas han encontrado el personaje que inicia todos los sucesos de la misma y que 3 personas no la han podido localizar. Por tanto, esta historia está peor situada que las dos anteriores.

¿Has jugado esta historia?

10 respuestas

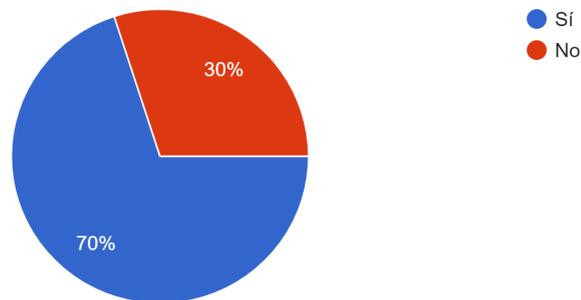


Figura 41: Porcentaje de jugadores que han jugado la historia número tres.

El 70% ha jugado frente a un 30% que no.

Como ha ocurrido con la historia anterior, ninguno de los jugadores que ha jugado esta historia la identifica como la narrativa principal, por lo que su diseño como narrativa secundaria se podría considerar correcto (figura 42).

¿Es la historia principal?

7 respuestas

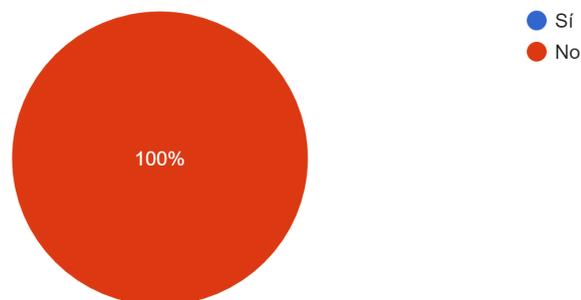


Figura 42: Porcentaje de jugadores que piensan que la historia número tres es la historia principal, siendo no para el 100% de los jugadores de la historia.

Sin embargo, al contrario que con la Historia 2, los jugadores no creen que pueda llegar a considerarse una historia principal. El resultado en este caso ha sido rotundo, teniendo al 100% de los jugadores que han interactuado con esta narrativa afirmando que no podría ser principal (figura 43). Los motivos son variados, pero el más repetido es que no tiene un avance bien definido. El segundo motivo es que hay demasiados personajes que además cuentan con poca importancia narrativa, por lo que se puede asumir que la cantidad de los personajes no influye en la identificación, si no que lo que puede llegar a ser relevante

es la propia relevancia de los personajes en la narrativa y cómo pueden llegar a afectar a la misma.

¿Crees que esta historia podría ser la principal?

7 respuestas

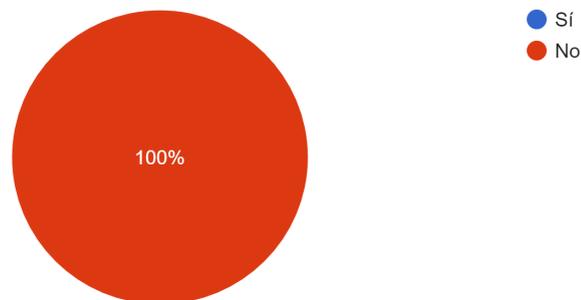


Figura 43: Porcentaje de jugadores que piensan que la historia número tres podría ser la historia principal. El 88,9% piensa que sí frente a un 11,1% que piensa lo contrario.

Según los datos obtenidos, representados en la figura 44, no tiene una gran duración, obteniendo su máximo valor de 3 personas en el punto 1 y confirmando el resultado anterior en el que se comentaba que la historia no avanzaba mucho, considerando que es demasiado corta como para representar una narrativa fuerte.

¿Te ha parecido muy larga?

7 respuestas

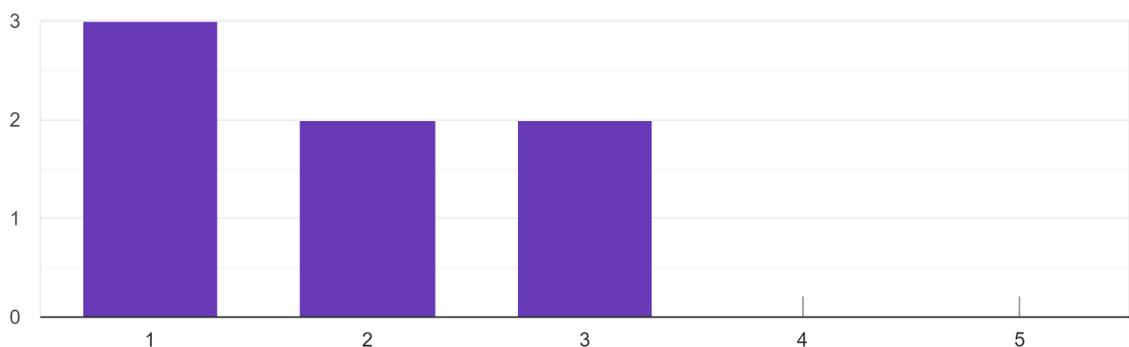


Figura 44: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

En cuanto a la cantidad de personajes, como se ha comentado anteriormente según los comentarios de algunos jugadores, intervienen demasiados, con un valor máximo de 3 personas en el punto 4 (figura 45).

¿Hay pocos o demasiados personajes involucrados?

7 respuestas

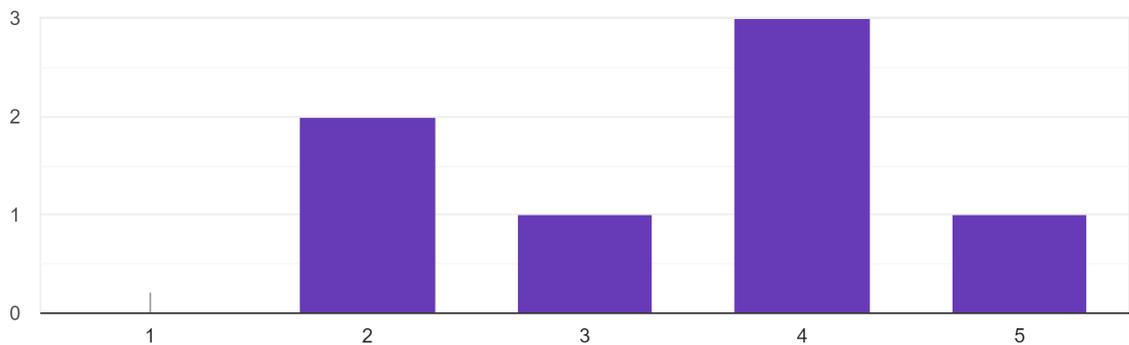


Figura 45: Porcentaje de jugadores que valoran la cantidad de personajes de la historia, siendo el valor 1 pocos personajes y 5 muchos personajes.

Observando los resultados generales obtenidos en esta historia, queda reflejado que si la duración es corta y tiene demasiados personajes, la historia tiende a considerarse como una secundaria.

Analizando más los datos, vemos que la importancia narrativa de los personajes, a pesar de contar con muchos, no dan una información muy relevante y tampoco son personajes que destaquen en el desarrollo de la trama. Si además tenemos en cuenta las observaciones de las historias anteriores, vemos que los jugadores prestan más atención a la información y a la importancia dentro de la historia y el mundo de los personajes antes que la cantidad que aparecen en ella.

Esto luego será relevante a la hora de modificar la hipótesis, ya que se cambia la cantidad de los personajes por la relevancia de los personajes en el mundo y la trama de la historia.

6.1.4 Historia 4

La última historia trata la desaparición del padre de un niño que se puede encontrar cerca de una casa en Fyrst. Tras encontrar al padre muerto en el bosque de Dara, se descubre que tenía una carta que darle al hijo y hay que volver para dársela e informarle acerca de la muerte de su padre.

La última historia ha sido jugada por el 80% de los jugadores, es decir 8 personas, como se puede ver en la figura 46. Sigue manteniendo un alto número de jugadores que han interactuado con dicha historia, siendo la segunda historia secundaria más jugada de las tres.

¿Has jugado esta historia?
10 respuestas

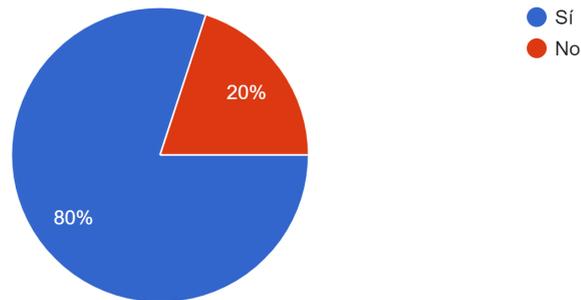


Figura 46: Porcentaje de jugadores que han jugado la historia número cuatro. El 80% ha jugado frente a un 20% que no.

A la hora de clasificar esta historia, 7 jugadores que quedan reflejados en el 87,5% del total indican que la historia no es principal, por lo que al igual que las otras se interpreta como secundaria (figura 47).

¿Es la historia principal?
8 respuestas

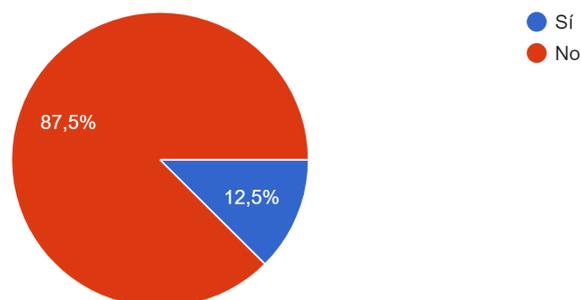


Figura 47: Porcentaje de jugadores que piensan que la historia número cuatro es la historia principal. El 12,5% piensa que sí frente a un 87,5% que piensa lo contrario.

Esta historia, a pesar de no ser considerada como principal, si tiene más potencial para convertirse en una en comparación con las anteriores consideradas como secundarias,

con un total de 3 jugadores que representan el 37,5% y que podrían interpretarla como principal (figura 48).

¿Crees que esta historia podría ser la principal?
8 respuestas

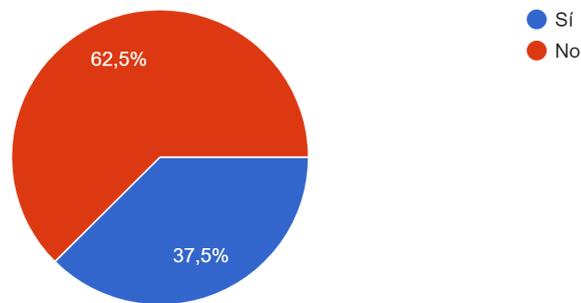


Figura 48: Porcentaje de jugadores que piensan que la historia número cuatro podría ser la historia principal. El 37,5% piensa que sí frente a un 62,5% que piensa lo contrario.

Los jugadores argumentan esa respuesta con una posible continuidad de la historia, con una trama con final abierto que sorprende a los jugadores y que finaliza con la aparición de un objeto que puede desarrollarse posteriormente, además de poder desarrollar la trama alrededor del niño que inicia la misión o el padre, que está envuelto en problemas con una organización de soldados y es algo que los jugadores han remarcado en algunos de los comentarios.

En cuanto a la duración tenemos una duración media-corta, con un máximo de 4 personas en el punto 2 (figura 49). Esto puede significar que la historia es lo suficientemente corta como para no considerarla principal, dando por cierta una de las variables de la hipótesis.

¿Te ha parecido muy larga?

8 respuestas

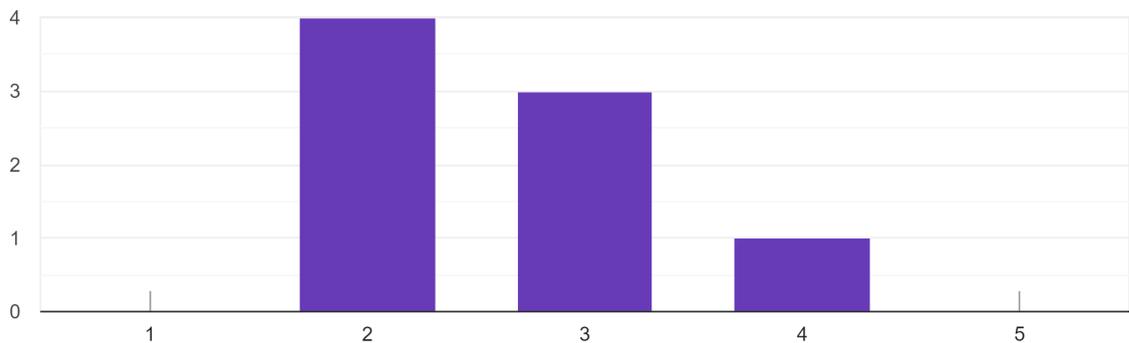


Figura 49: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

La aparición de los personajes también se ve escasa, con un máximo de 5 personas en el punto 2 (figura 50). Sin embargo, que varias personas hayan indicado que esta historia podría llegar a ser la principal, refuerza la idea mencionada anteriormente de que lo que influye en cuanto a los personajes es la importancia de dichos personajes en la trama y en el mundo y no la cantidad numérica de personajes sin más.

¿Hay pocos o demasiados personajes involucrados?

8 respuestas

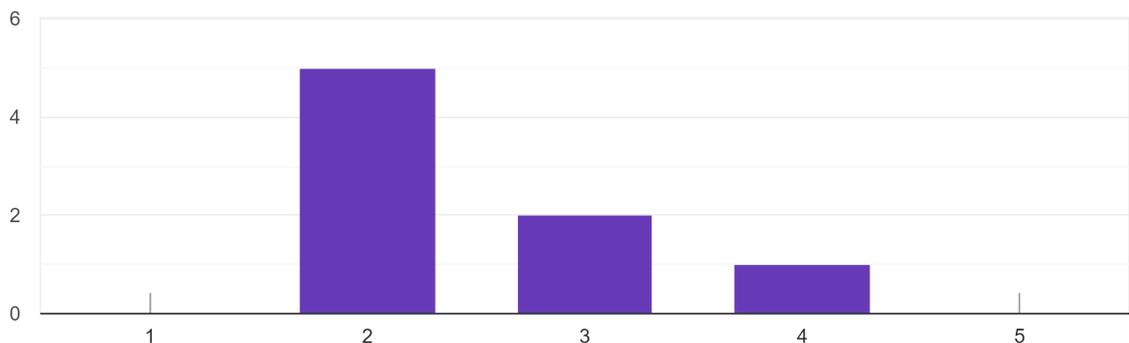


Figura 50: Porcentaje de jugadores que valoran la cantidad de personajes de la historia, siendo el valor 1 pocos personajes y 5 muchos personajes.

6.2 Discusión de los resultados de la experimentación

Tras obtener los resultados de las encuestas y habiendo realizado un análisis de los datos, se ha podido comprobar que muchas de las variables que componían la hipótesis de partida no eran del todo correctas y necesitaban una revisión para funcionar del todo en un sistema de clasificación automatizado.

Para empezar, la cantidad de personajes no influye directamente de forma determinante en la detección de historias y narrativas secundarias. Sin embargo, se ha encontrado nueva información que podría ser útil a la hora de contar con los personajes en la clasificación y es la importancia narrativa de los personajes. Esta se podría dividir en dos tipos para obtener un dato más preciso: importancia narrativa e importancia diegética.

La primera de las dos se centraría en la relevancia que tienen los personajes en la historia en la que participan. Esto es, cuánto influyen en la trama y qué papel desempeñan en la misma. Por ejemplo, un personaje que va a acompañar al jugador a lo largo de toda la narrativa tendría una importancia muy alta, ya que su influencia en la trama es bastante alta. Sin embargo, no sería más alta que la importancia del villano, cuyo impacto es mucho mayor en la trama. Así que, si se estableciera un rango de importancia del 1 al 100, siendo 100 el más importante y 1 un personaje que no influye de ninguna manera, el acompañante tendría una importancia de 80, por ejemplo, y el villano tendría una importancia de 95. Ambos son valores muy altos pero sigue existiendo una diferencia entre las relevancias.

La segunda es un poco más compleja, ya que es la importancia que tendría el personaje dentro del mundo, qué papel desempeña para con la diégesis del videojuego. En el caso propuesto con anterioridad, los personajes tendrían una importancia diferente, ya que sus papeles en el mundo pueden variar. Por ejemplo, suponiendo que el acompañante es un herrero de profesión, su importancia sería bastante baja. Mientras que el villano podría ser una figura importante como un rey, un noble o un hechicero de alto rango, siguiendo el ejemplo de la fantasía. En este caso sus importancias diegéticas tendrían unos valores mucho más separados si se pusieran en una escala similar, del 1 al 100.

Por tanto, se descarta el planteamiento de la variable centrada en la cantidad de personajes y se sustituye por la relevancia de los mismos, añadiendo un sistema de

puntuación centrado en la importancia narrativa y diegética, haciendo una media ponderada de ambas, dando lugar a la relevancia del personaje respecto al total del juego.

A su vez, la longitud se ve afectada por esta relevancia de personajes, algo que en un primer momento no se planteó que estuvieran relacionados. Por algunos de los datos recopilados, se percibe la duración de una historia como más larga o más corta por contar con objetos clave o personajes narrativamente más enriquecidos. Además de que la medida por capítulos no es tan perceptible como la medida por escenas, como se ha podido apreciar en los datos, ya que las escenas se encuentran en lugares claramente diferenciados y los capítulos son algo más abstractos, en cuanto a la percepción del jugador. Las dos primeras historias, pese a tener la misma longitud en capítulos han sido percibidas de maneras diametralmente opuestas, siendo la primera identificada como principal y la segunda como secundaria de manera casi rotunda. Por tanto, la opción lógica es cambiar esa variable de longitud en capítulos por longitud en escenas para que la hipótesis sea más precisa.

Por otro lado, la cantidad de lugares visitados sí influye a la hora de entender las historias como principales o como secundarias, ya que se ha tenido en cuenta a la hora de discriminarlas e incluso ha sido mencionado como determinante en algunas respuestas.

Cabe destacar que el orden en el que se distribuyen las historias también influye a la hora de que los jugadores la entiendan como principal o no. Si una historia aparece primero en el juego antes que otras, esa suele ser percibida como principal antes que las demás.

Capítulo 7: Aplicación de cambios tras las pruebas

Para comprobar si el sistema está diseñado para analizar la hipótesis una vez redefinida, se realiza un estudio con las variables definidas en dicha hipótesis y comprobar si es válida. En primer lugar, se almacena la información concreta de cada historia:

La primera historia, considerada como la historia principal, cuenta con 3 capítulos entre los que se reparten un total de 7 escenas. En ellas encontramos la intervención de 3 personajes que dotan la historia de una relevancia de 119 puntos, siendo la suma de las relevancias de sus personajes de 20, 49 y 50 respectivamente. También cuenta un total de 8 escenarios diferentes por lo que conduce la trama.

La segunda historia, enfocada a la longitud de esta, cuenta también con 3 capítulos, pero esta vez teniendo un total de 6 escenas. En ella se encuentran 4 personajes con una relevancia de 7, 15, 10 y 10, que ofrece a la historia una relevancia de 42 puntos. En cuanto a los lugares, se recorre un total de 9 lugares diferentes.

La tercera historia, que cuenta con la intervención de muchos personajes, está formada por 2 capítulos, compuestos en conjunto por 4 escenas. Analizando la variable correspondiente, esta historia cuenta con 9 personajes de relevancia 15, 10, 5, 18, 25, 30, 10, 34 y 20, que suman una relevancia total de 167 puntos. Esta historia se desarrolla en 6 escenarios distintos.

Por último, la cuarta centrada en el número de lugares visitados, cuenta con un único capítulo que contiene 5 escenas. En estas escenas se interactúa con 4 personajes de relevancia 5, 32, 20 y 45, con un valor de 102 puntos de relevancia total. Como esta historia se centra en los lugares, la trama lleva a recorrer un total de 7 lugares diferentes.

Una vez recopilada la información de cada capítulo, se calculan los valores totales de cada variable. Las historias en conjunto cuentan con un total de 9 capítulos, desarrollados en 22 escenas. Estas escenas se desarrollan en un total de 14 lugares donde se encuentran repartidos los diferentes personajes que en total, dotan al conjunto de historias de una relevancia de 447 puntos.

Con estos valores, se realizan las operaciones que ofrecen el valor, calculado en porcentajes, de la importancia de cada una de las variables y que se utilizan para calcular la media de importancia de una historia desarrollada en el **Capítulo 4.4 Segunda versión del motor**. Como ejemplo, se pueden ver los cálculos que se realizan en el prototipo de juego en la figura 51.

Historia	Capítulos (%)	Escenas (%)	Rel. de Personajes (%)	Lugares (%)
1	33	31,8	26,6	50
2	33	27,27	9,39	42,85
3	22,2	18,18	37,3	28,5
4	11,1	22,7	22,8	28,5

Figura 51: Cálculo en porcentajes de cada uno de los valores que intervienen en el cálculo de la relevancia de una historia.

Para calcular las medias, se analizan tres opciones diferentes para analizar la longitud de una historia, recogidas en la figura 52. En la primera opción, se analiza la longitud de la historia empleando tanto la cantidad de capítulos como la cantidad de escenas, de esta forma se tiene en cuenta la relación entre las dos cantidades. Como segunda opción, se analiza manteniendo solo la cantidad de capítulos, como establece el planteamiento de la hipótesis. Por último, se calcula la media empleando solo las escenas, ya que los jugadores perciben mejor el cambio entre escenas que el cambio de los capítulos.

Media con capítulos y escenas (%)	Media con escenas (%)	Media con capítulos (%)
35.35	36.13	36.53
28.12	26.503	28.413
26.54	27.993	29.33
21.27	24.66	20.8

Figura 52: Cálculo en porcentajes de la relevancia de una historia empleando diferentes opciones que analizan la longitud de dichas historias de diferentes maneras.

Tras este estudio y en análisis de los resultados en el **Capítulo 6: Pruebas del prototipo**, se establece el cálculo de la longitud de la historia mediante el número de escenas que la forman por la percepción de los jugadores y por la diferencia que establece entre las historias comparando las tres formas de calcular la media estudiadas.

Capítulo 8: Pruebas finales

Para poder probar que el motor computacional funciona correctamente, se ha realizado una prueba final con otro grupo de personas con tal de obtener datos acerca del funcionamiento del motor y el algoritmo de clasificación de las historias.

Tras realizar los cambios mencionados anteriormente, se procedió a realizar la prueba final, esta vez con el motor ya implementado y funcionando con el juego, proporcionando historias según se iban necesitando o entraban dentro del lugar en el que estaba el jugador en cada momento.

El objetivo de esta prueba es determinar si la clasificación realizada por el motor es la correcta y si la situación de las historias también, ya que después de haber aplicado los cambios no se encuentran todas las historias disponibles desde un primer momento. Para ello se volvió a generar un ejecutable del prototipo que incluía el motor con el algoritmo ya implementado. Este se volvió a subir a itch.io¹⁵ y después se proporcionó una nueva encuesta para que los jugadores la contestaran tras probar el prototipo.

En este cuestionario se preguntaba a los jugadores de nuevo qué historias habían jugado y cuánto habían tardado en completarlas. Tras esto se les preguntaba por cada una de las historias y cómo las habían detectado ellos, principales o secundarias, y por qué. Finalmente se informaba de la clasificación que había hecho el motor historia a historia y se preguntaba por qué variable había sido relevante para obtener los resultados que se habían obtenido y de esta manera observar si coincidían con los resultados del motor o si se encontraba algún detalle más a tener en cuenta a la hora de clasificar las historias y que no se había tenido en cuenta hasta ahora.

Esta vez se consiguió que 14 personas probaran el prototipo y rellenaran la encuesta que lo acompañaba con tal de recabar datos del funcionamiento del motor y del algoritmo implementado para ordenar y clasificar las historias. Con tal de comparar los resultados obtenidos con el anterior experimento, se han usado las mismas historias. Sin embargo, estas historias aparecían según lo dictaba el motor, colocando personajes no jugables e historias según se iban necesitando y no como estaban colocadas la primera vez.

¹⁵ <https://itch.io/>

En la siguiente sección se analizará los datos obtenidos por cada historia individualmente, comparándolos en algunos aspectos con los resultados obtenidos en la prueba anterior y poniendo especial énfasis en el funcionamiento del algoritmo y cómo ha sido percibido por los jugadores.

8.1 Resultados

Tras analizar los resultados más generales de todas las historias, se han podido llegar a nuevas conclusiones que pueden aportar más variables y parámetros para ajustar la hipótesis y el algoritmo de manera que la clasificación y detección de historias sea más precisa.

Uno de los datos que es bastante relevante a la hora de la identificación de historias es que los comentarios pequeños o sin importancia, como puede llegar a hacer algún personaje no jugable en un pueblo o una ruta entre pueblos, pueden dar a entender que hay una historia detrás de ellos. Esto es porque muchas personas han identificado un mayor número de historias de las que hay realmente en el juego por los diálogos genéricos que hay implementados en algunos puntos del mapa.

En la figura 53 se puede observar que un gran número de personas han identificado mínimo 5 historias, una más de las que hay realmente. Este dato puede parecer poco relevante y tomarse como margen de error. Sin embargo, al analizar los comentarios acerca de las descripciones de las historias identificadas, muchas personas han identificado un diálogo genérico como el inicio de una posible historia. Este es el comentario que hace un personaje aleatorio en referencia a un reloj que ha perdido. La mayoría de las respuestas estaban justificadas con que la historia involucraba un objeto y también se ha destacado al personaje que realiza el comentario como uno relevante a la hora de identificar la historia.

¿Cuántas historias crees que hay en el juego?

14 respuestas

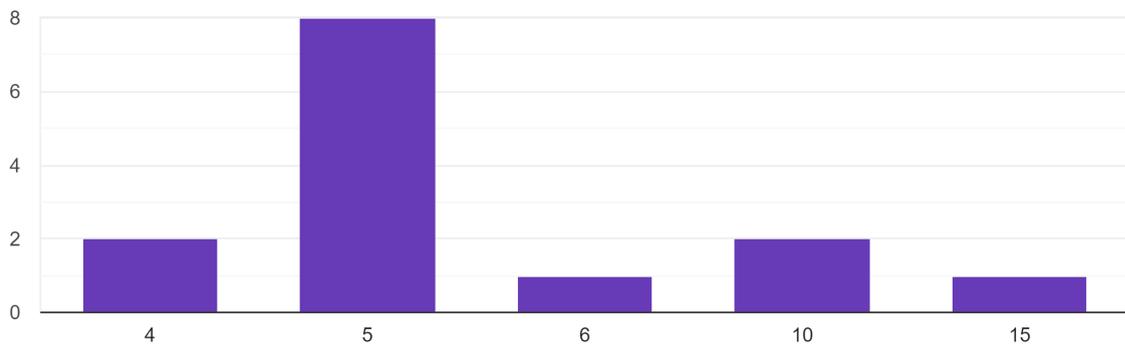


Figura 53: Número de historias identificadas por los jugadores en la sesión de juego realizada.

Por tanto, aunque sea una historia pequeña, este tipo de diálogos sin importancia y que simplemente están ahí para dar más profundidad a la diégesis del videojuego o al mundo para que el jugador lo perciba como un lugar vivo y real pueden llevar a los jugadores a buscar una historia relacionada con ellos. No obstante, cabe destacar que todos los jugadores identificaron esta historia como una secundaria, llevando a algunas conclusiones que se explican en **Capítulo 9: Discusión** y **Capítulo 10: Conclusiones y trabajo futuro**.

Sin embargo esto no ha sido un factor limitante para completar las historias que estaban disponibles para jugar a lo largo de toda la sesión. Aproximadamente el 50% de los usuarios ha podido completar todas o casi todas las historias del juego, completando un total de 3 historias un 28,6% de los usuarios, 4 de 14 totales, y terminando todas las historias un 21,4% de los jugadores, 3 de 14 totales. Estos datos muestran que las historias se habían colocado correctamente y que era posible completar todas en una sola sesión o al menos un porcentaje bastante alto.

Por otro lado, también hay que destacar que el 28,5% de los jugadores sólo han completado 2 o menos historias de las que había disponibles en el juego, por lo que habría que estudiar la correlación entre el tiempo invertido en la sesión de juego y el porcentaje de historias completadas, pero eso no se ha tenido en cuenta en este trabajo.

Otro dato que también es relevante destacar es que el 100% de los jugadores ha jugado y completado la historia principal del juego. Sin embargo, sólo el 78,6% considera

que es la historia principal, algo que se puede deber a la situación de los personajes o la longitud de la misma historia. Esto se analiza más en detalle en la siguiente sección.

8.1.1 Historia 1

Esta historia es la que comienza en el pueblo de Fyrst y se debe hablar con el alcalde para que envíe al jugador a entregar un cáliz en el pueblo de Dara. Esta historia se diseñó para ser la principal y es la que el motor también ha identificado como principal en esta prueba.

Como se puede observar en el gráfico correspondiente con la figura 54 todos los jugadores han jugado esta historia. Esto quiere decir que de las 14 personas que han participado en la prueba, todas han podido jugar esta narrativa. Esto quiere decir que estaba bien situada y era accesible por todos los jugadores.

¿Has jugado esta historia?
14 respuestas

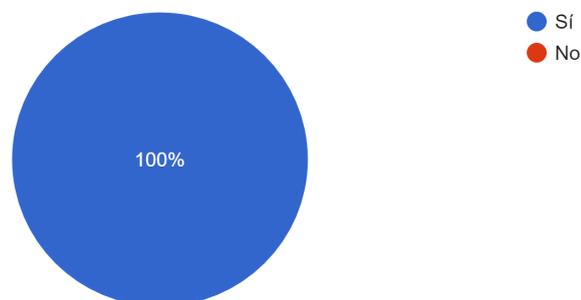


Figura 54: Porcentaje de jugadores que han jugado la historia número uno. Todos los participantes han jugado esta historia

Se puede observar en la figura 55 que un porcentaje muy alto de los jugadores han identificado esta historia como la principal. Entre las razones para identificar esta historia como la principal las más destacables son las de situación de la historia, personajes y objetos involucrados en la misma.

¿Crees que esta historia es la principal?

14 respuestas

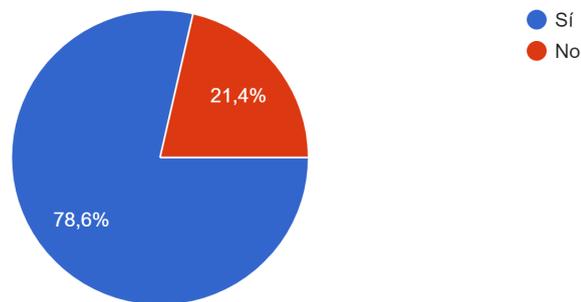


Figura 55: Porcentaje de jugadores que piensan que la historia número uno es la historia principal. El 78,6% piensa que sí frente a un 21,4% que piensa lo contrario.

Muchos jugadores han destacado que esta historia es principal porque es la primera que aparece en el juego y la primera a la que se juega. Esto señala que el orden en el que se presentan las historias es relevante a la hora de decidir cuál es la más importante o principal. Por tanto, el lugar en el que están colocados los personajes no jugables o los objetos que inicien una historia es importante también.

En la figura 56 se puede observar la percepción de la longitud de la historia que han tenido los diferentes jugadores. Como se puede observar, ha habido una gran percepción de la longitud de la historia como una historia corta aunque los valores se aproximan más a una duración media si se cuenta con los valores más altos de duración, ya que nadie ha percibido la historia como muy corta.

¿Te ha parecido muy larga?

14 respuestas

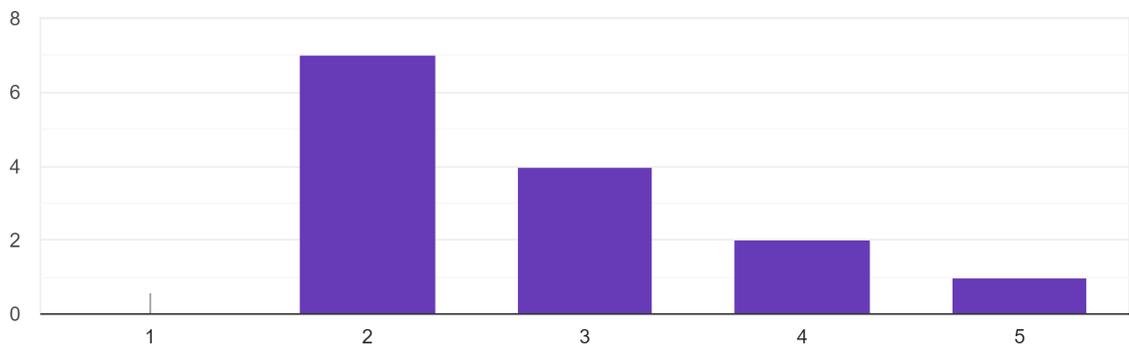


Figura 56: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

Por otro lado, la percepción de los lugares sí se corresponde más con las respuestas a la primera pregunta planteada en esta sección. Los resultados dan una percepción de lugares visitados media-alta, teniendo el máximo en 3 como una cantidad de lugares adecuados pero contando con valores muy altos en 4 y 5 indicando una cantidad alta de lugares visitados (figura 57).

¿Cuántos lugares crees que has visitado?

14 respuestas

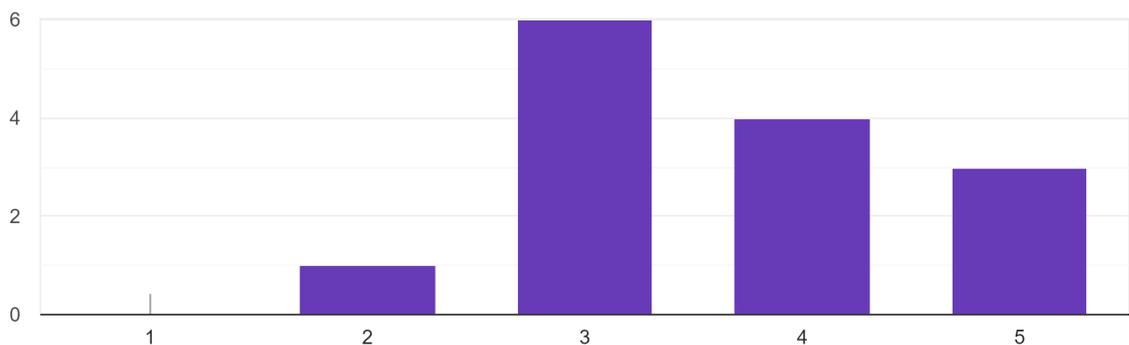


Figura 57: Porcentaje de jugadores que valoran el número de lugares recorridos en la historia, siendo el valor 1 equivalente a pocos lugares y 5 una muchos.

Por último, la valoración de la importancia de los personajes ha dado lugar a unos datos que reafirman la teoría de que la importancia de los personajes que aparecen en una narrativa es muy importante a la hora de diferenciar entre una narrativa principal y una secundaria. Con un máximo en 4 con 6 personas opinando que los personajes son

importantes y que dan información relevante para la trama y con 4 personas votando el valor 5, dando a entender que los personajes han sido muy relevantes y de mucha importancia (figura 58).

¿Crees que los personajes involucrados son importantes o dan información importante en la historia?

14 respuestas

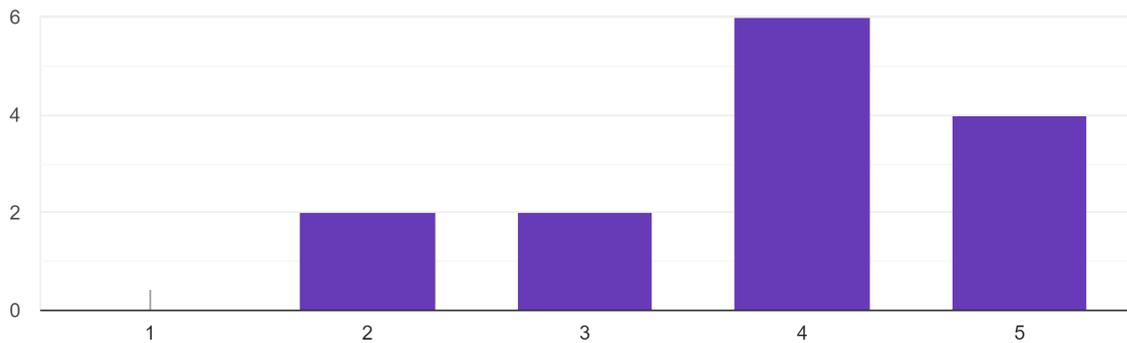


Figura 58: Porcentaje de jugadores que valoran la relevancia de los personajes involucrados en la historia, siendo el valor 1 poca relevancia y 5 una gran relevancia.

Por último, en la figura 59 se puede observar lo que los jugadores creen que el motor ha tomado en cuenta a la hora de clasificar la historia y determinar que se trata de una narrativa principal. El parámetro que más destaca, con un total de 10 votos es la cantidad de personajes importantes, confirmando que el parámetro más relevante a la hora de ordenar y clasificar narrativas es la relevancia de los personajes dentro de la trama y la diégesis del juego.

El segundo parámetro más votado es la cantidad de lugares visitados en la narrativa, con un total de 9 votos, afirmando que la cantidad de los lugares por los que transcurre una narrativa provoca que pueda ser más importante que las demás.

Por otro lado, el parámetro de la duración de la historia es el menos votado de los establecidos en la hipótesis de partida, contando sólo con 6 votos. Este dato muestra que la duración de una historia, aun siendo relevante a la hora de establecer la importancia de una narrativa, no es el valor más importante a tener en cuenta para realizar la clasificación, dando mayor importancia a los personajes involucrados y los lugares visitados.

Finalmente, cabe destacar los últimos comentarios, que no son parte de los proporcionados en la encuesta. Estos señalan la sensación que transmite la historia, de ser algo más que simplemente entregarle algo a alguien como un recado y la mención al emperador, un personaje que si bien no se ha tenido en cuenta dentro de la historia tal como se ha planteado para este prototipo, es un personaje importante dentro de la diégesis y eso da sensación de contener una historia mucho más extensa más allá de lo que ya se muestra. Esto confirma lo planteado para la importancia de los personajes observando estos datos. La relevancia e importancia de un personaje tanto narrativa como diegética es un factor relevante a la hora de clasificar las narrativas en secundarias y principales.

El motor ha considerado que esta historia es PRINCIPAL, ¿a qué crees que se debe esta clasificación? (Puedes añadir otra si crees que afecta algo más)

14 respuestas

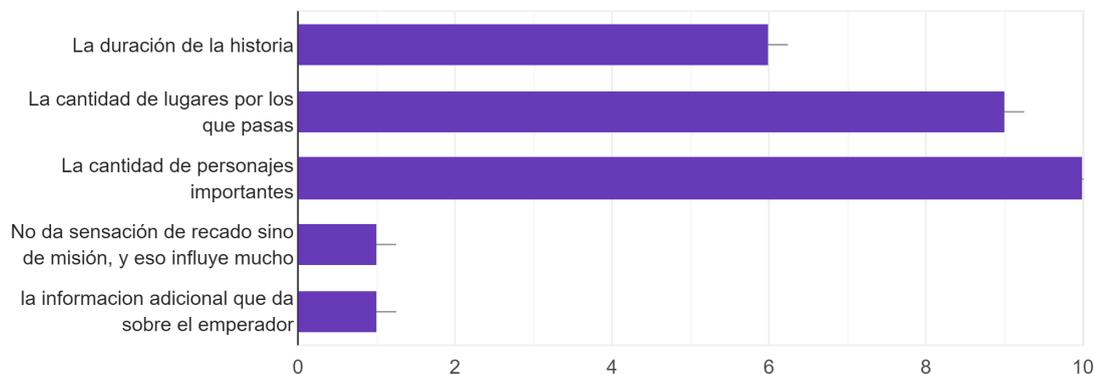


Figura 59: Porcentaje de jugadores que valoran las variables definidas en la hipótesis y algunas variables no estudiadas.

8.1.2 Historia 2

Esta es la historia en la que un hombre pide al jugador que busque a su nieta porque se fue a buscar una espada mágica y todavía no ha vuelto a casa. En esta historia se explora tanto el pueblo de Dara como Fyrst y sus alrededores, así como la mazmorra y la cueva que se encuentran en el camino que une Fyrst con Dara.

En esta fase de pruebas han jugado esta historia un total de 12 personas, un 85,7% del total de personas que han realizado la prueba (figura 60). Aunque no todos los jugadores han podido jugar esta historia, se puede observar que un gran número de personas han podido encontrar y jugar esta narrativa, por lo que la situación de la misma sigue siendo correcta.

¿Has jugado esta historia?

14 respuestas

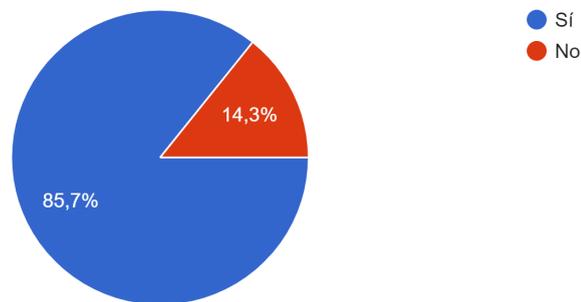


Figura 60: Porcentaje de jugadores que han jugado la historia número dos. El 85,7% ha jugado esta historia frente a un 14,3% que no.

Sin embargo, comparada con la Historia 1, la ha jugado menos gente. Esto puede indicar que aunque la situación es correcta para una historia secundaria, no siempre es accesible para todos los jugadores.

Por otro lado, como se puede observar en la Figura 61, ninguno de los jugadores que han jugado esta historia opina que sea una narrativa principal. En los comentarios que han realizado los jugadores para complementar esta información, se ha destacado mucho que no es una narrativa relevante y que no aporta mucho al desarrollo del juego en general. Además, también se ha mencionado que no se llega hasta ella siendo guiado por el primer NPC con el que se interactúa, afirmando la idea de que las narrativas también ganan relevancia por su orden de aparición en el juego.

¿Crees que esta historia es la principal?

12 respuestas

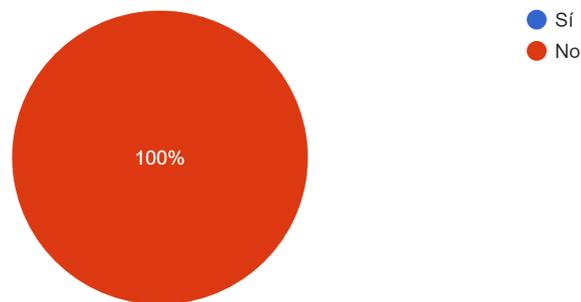


Figura 61: Porcentaje de jugadores que piensan que la historia número dos es la historia principal.

Todos los jugadores que han interactuado con esta historia afirma que no.

En la Figura 62 se puede observar que los jugadores han percibido esta historia de una duración media-corta, con dos máximos en nivel de duración 2 y 3. Esto puede confirmar la idea de que la importancia de ciertos personajes influye en la percepción de la duración de las historias.

¿Te ha parecido muy larga?

12 respuestas

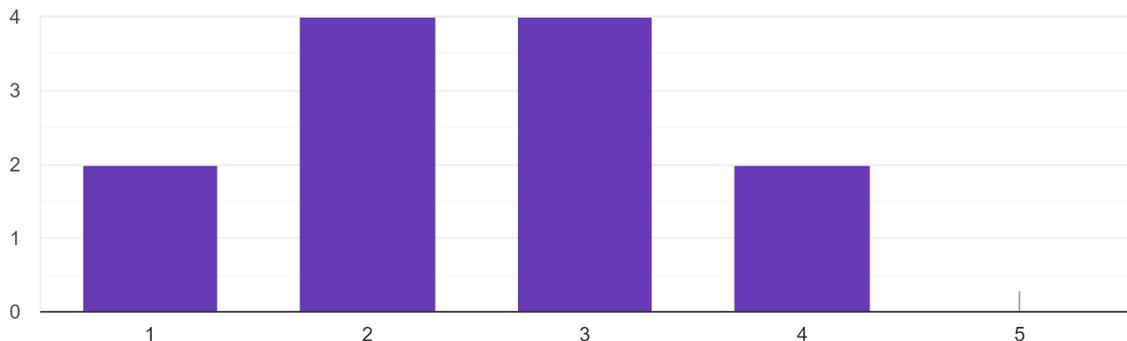


Figura 62: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

Sin embargo, la percepción de los lugares visitados se ha mantenido bastante normal, teniendo un máximo en el valor 3 de duración con 6 personas opinando que la cantidad de lugares visitados es media, como se puede ver en la figura 63.

¿Cuántos lugares crees que has visitado?

11 respuestas

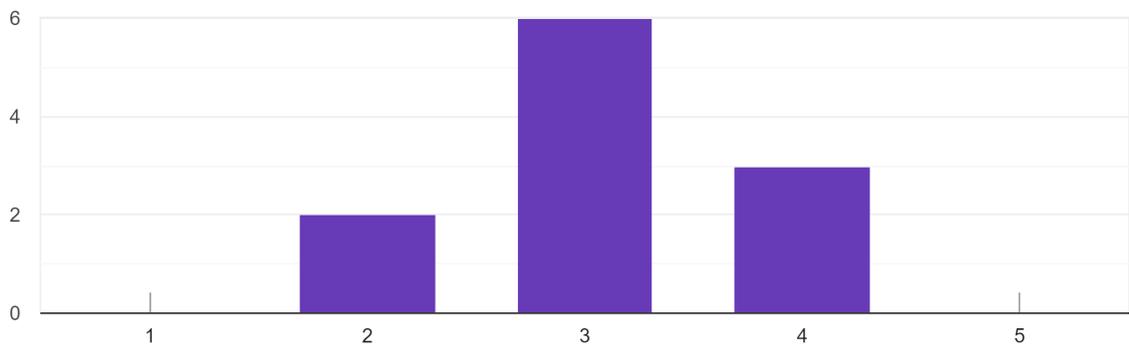


Figura 63: Porcentaje de jugadores que valoran el número de lugares recorridos en la historia, siendo el valor 1 equivalente a pocos lugares y 5 una muchos.

Por otro lado, la percepción de la importancia de los personajes de la historia ha resultado en unos valores muy bajos, con un máximo en 3, teniendo una importancia media, pero muchas personas opinando que la importancia de los personajes es baja (figura 64).

¿Crees que los personajes involucrados son importantes o dan información importante en la historia?

12 respuestas

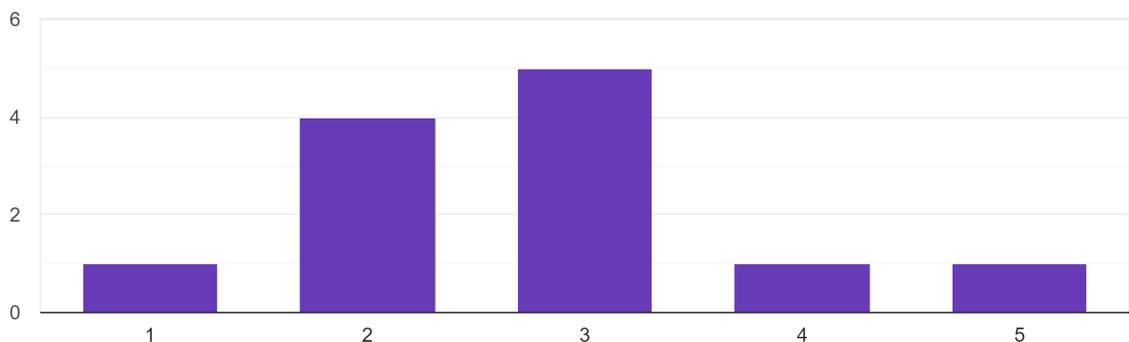


Figura 64: Porcentaje de jugadores que valoran la relevancia de los personajes involucrados en la historia, siendo el valor 1 poca relevancia y 5 una gran relevancia.

Por último, planteando la clasificación que había proporcionado el motor de esta historia y preguntando por qué había sido esta según se puede ver en la figura 65, la importancia de los personajes ha sido la más relevante con un total de 9 votos. Esto confirma la teoría analizada en la anterior historia de que la relevancia de los personajes es

mucho más importante en la detección de narrativas principales de lo que parece a simple vista.

El segundo parámetro más votado es la duración de la historia con 5 votos, que, relacionándolo con los 2 valores que no estaban previstos, también se ve influida por la importancia de los personajes. Por tanto, que estos dos valores sean tan altos indica que ambos están relacionados directamente. A mayor relevancia de los personajes, mayor percepción de la longitud de la historia y mayores opciones de ser la narrativa principal del juego.

Por último mencionar los valores que sólo han obtenido 1 voto y que el que más destaca es la cantidad de lugares visitados en la historia. Este valor ha obtenido una puntuación muy baja, por lo que no ha sido muy relevante para los jugadores a la hora de determinar si esta historia era secundaria o principal.

Las otras dos respuestas restantes hablan de la continuidad de la historia y de que no parece una misión muy relevante o que vaya a aportar algo a la trama completa del videojuego. Aunque se menciona que Julie, la persona a la que hay que buscar en la cueva podría llegar a convertirse en un personaje pasivo en la historia.

El motor ha considerado que esta historia es SECUNDARIA, ¿a qué crees que se debe esta clasificación? (Puedes añadir otra si crees que afecta algo más)
12 respuestas

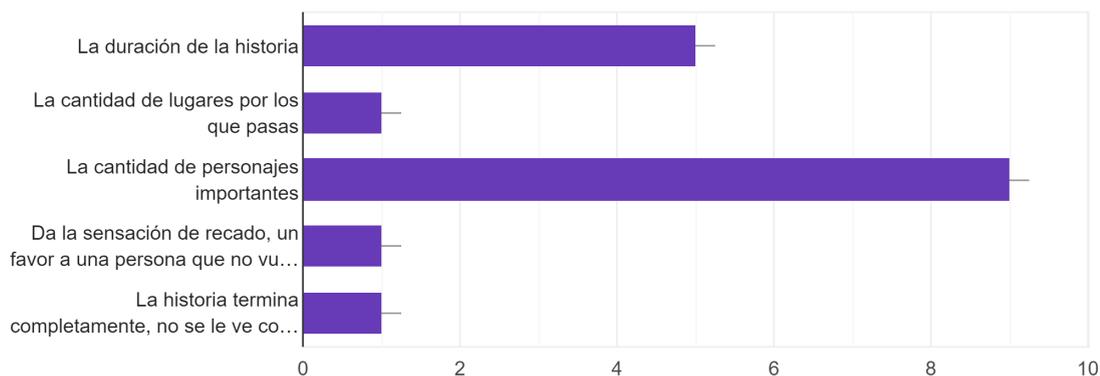


Figura 65: Porcentaje de jugadores que valoran las variables definidas en la hipótesis y algunas variables no estudiadas.

8.1.3 Historia 3

Esta historia transcurre entre Fyrst y la taberna de Dara, donde hay que hablar con una serie de personajes para que se unan al equipo con tal de buscar un objeto que se ha perdido en el bosque de Dara. Esta ha sido la historia menos jugada de esta fase de pruebas.

Como se puede observar en la Figura 66, tan sólo el 50% de los jugadores, un total de 7, ha jugado esta historia. Esto indica que no estaba bien situada y era difícil de localizar para los jugadores.

¿Has jugado esta historia?
14 respuestas

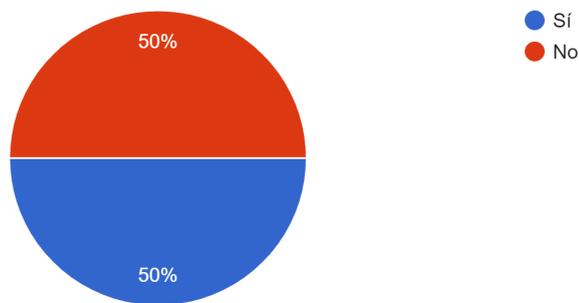


Figura 66: Porcentaje de jugadores que han jugado la historia número tres. El 50% ha jugado esta historia frente a un 50% que no.

De estas 7 personas que han jugado la historia, 6 de ellas piensan que no podría ser la principal, frente a una de ellas que cree que sí puede serlo porque se reúne a un equipo de personas y se va en busca de un objeto mágico, lo que puede desembocar en una historia más extensa (figura 67).

¿Crees que esta historia podría ser la principal?

7 respuestas

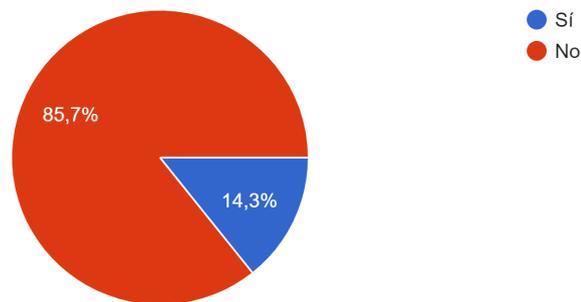


Figura 67: Porcentaje de jugadores que piensan que la historia número tres es la historia principal. El 14,3% piensa que sí frente a un 85,7% que piensa lo contrario.

En cuanto a la duración, como se puede observar en la Figura 68, se ha percibido como una historia demasiado larga. Esto se puede deber a la cantidad de personajes con los que hay que hablar para hacer que la historia avance. Esto sigue confirmando que los personajes influyen en la percepción de la duración de la historia.

¿Te ha parecido muy larga?

7 respuestas

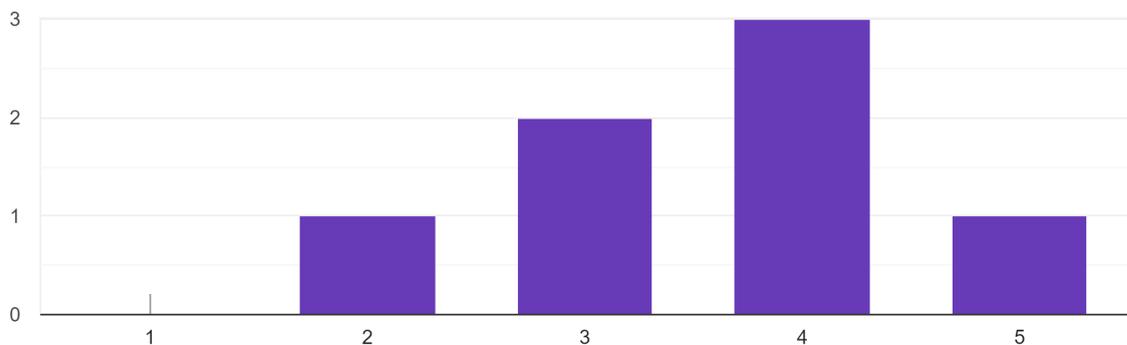


Figura 68: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

Por otro lado, la cantidad de lugares que se ha percibido como visitada es de muy pocos lugares visitados como se puede ver en la figura 69, con un máximo en el valor 2 con 3 personas opinando que se han visitado pocos lugares. Aunque los datos están repartidos

entre cantidad media y muchos, se puede observar un crecimiento hacia los valores más bajos.

¿Cuántos lugares crees que has visitado?

7 respuestas

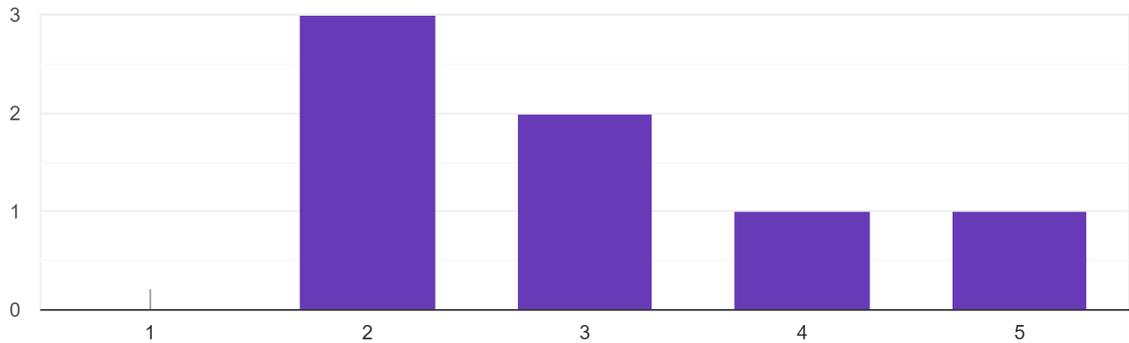


Figura 69: Porcentaje de jugadores que valoran el número de lugares recorridos en la historia, siendo el valor 1 equivalente a pocos lugares y 5 una muchos.

En cuanto a la relevancia de los personajes involucrados, los datos están muy repartidos entre valores medios, con tres máximos en los valores 2, 3 y 4 (figura 70). Esto puede indicar que la relevancia percibida de los personajes es algo difusa y cuesta dilucidar si los personajes que participan en esta historia son importantes o no.

¿Crees que los personajes involucrados son importantes o dan información importante en la historia?

7 respuestas

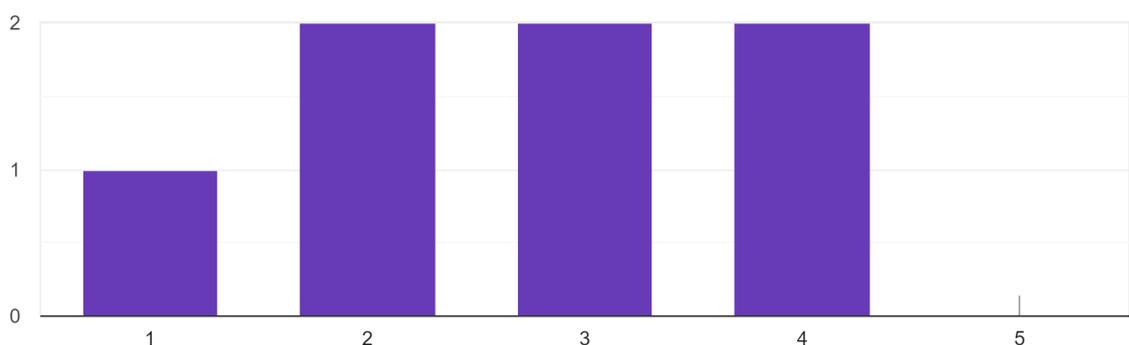


Figura 70: Porcentaje de jugadores que valoran la relevancia de los personajes involucrados en la historia, siendo el valor 1 poca relevancia y 5 una gran relevancia

Por último, la comparación con los datos obtenidos por la clasificación del motor son parecidos a los de la Historia 2, como se puede observar en la figura 71. La cantidad de

personajes sigue siendo la más relevante a la hora de clasificar las historias, seguida por la duración por lo que se puede establecer la misma relación que se planteaba en el análisis de la Historia 2 entre la percepción de longitud y los personajes involucrados en la trama y la narrativa.

La cantidad de lugares visitados en esta historia parece que no ha sido la más relevante de las tres variables a la hora de determinar si la historia es secundaria según los encuestados.

Por último, el último comentario que aparece hace mención a la cantidad de personajes y al posible potencial que estos presentan para crear una historia que pudiera llegar a ser principal, luego la cantidad de personajes influye, como se estableció desde un principio, pero vuelve a intervenir la relevancia de los mismos en la determinación de si una historia es principal o no.

El motor ha considerado que esta historia es SECUNDARIA, ¿a qué crees que se debe esta clasificación? (Puedes añadir otra si crees que afecta algo más)

7 respuestas

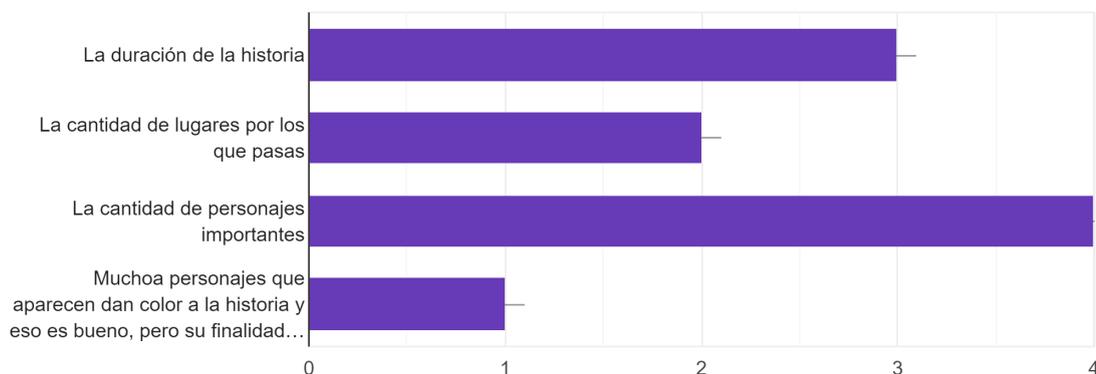


Figura 71: Porcentaje de jugadores que valoran las variables definidas en la hipótesis y algunas variables no estudiadas.

8.1.4 Historia 4

Esta historia es la que cuenta la historia de Kanda y su padre, que fue a realizar un recado en Dara y no volvió a Fyrst, por lo que Kanda se encuentra preocupado y pide ayuda al jugador para que encuentre a su padre y saber qué le ha pasado.

Esta historia ha sido jugada por más personas que la anterior, manteniéndose cerca del resto de historias en cuanto a personas que han encontrado esta narrativa, según se puede ver en la figura 72. Por tanto, se puede decir que la historia estaba bien situada y el motor la ha proporcionado correctamente al juego.

¿Has jugado esta historia?

14 respuestas

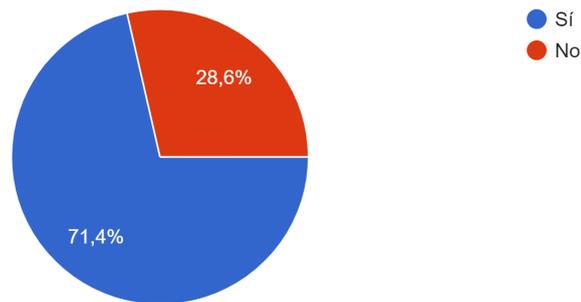


Figura 72: Porcentaje de jugadores que han jugado la historia número cuatro. El 71,4% ha jugado esta historia frente a un 28,6% que no.

Por otro lado, de los 10 jugadores que han jugado esta historia, solo uno de ellos cree que esta podría llegar a ser la narrativa principal del juego (figura 73). Comparándola con el resto de historias, obtiene el mismo resultado que la Historia 3, aunque esta contaba con menos personas que la hubieran jugado por lo que no son datos concluyentes. Esto indica que se podría haber percibido como una narrativa principal si algo hubiese cambiado, como la situación de la misma, ya que se menciona la importancia de algunos personajes involucrados y de los objetos que conducen la historia.

¿Crees que esta historia podría ser la principal?

10 respuestas

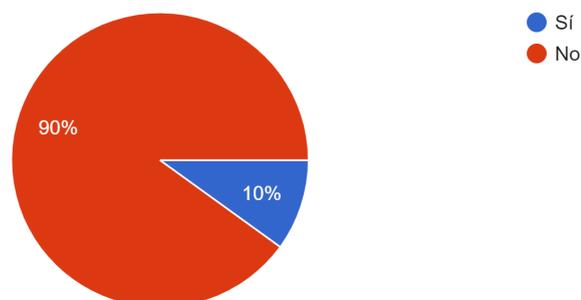


Figura 73: Porcentaje de jugadores que piensan que la historia número cuatro es la historia principal. El 10% piensa que sí frente a un 90% que piensa lo contrario.

Esta historia ha sido percibida en general con una duración media, aunque hay más votos en los valores más altos de duración, el máximo está en una duración de 3 con 4 votos (figura 74).

¿Te ha parecido muy larga?

10 respuestas

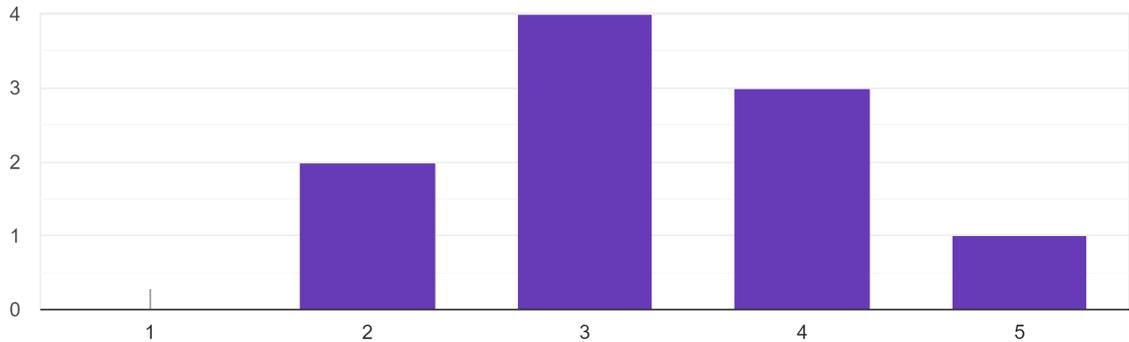


Figura 74: Porcentaje de jugadores que valoran la duración de la historia, siendo el valor 1 una duración corta y 5 una duración larga.

Por otro lado, los datos en relación con los lugares visitados durante la historia, como se puede observar en la Figura 75, son mucho más dispares. Aunque el máximo, de nuevo con 4 votos, se encuentra en 2 significando que la cantidad de los lugares es poca, los valores 4 y 5 también han obtenido una puntuación bastante alta, por lo que se podría deducir que la cantidad de lugares visitados ha obtenido un valor promedio como resultado final.

¿Cuántos lugares crees que has visitado?

10 respuestas

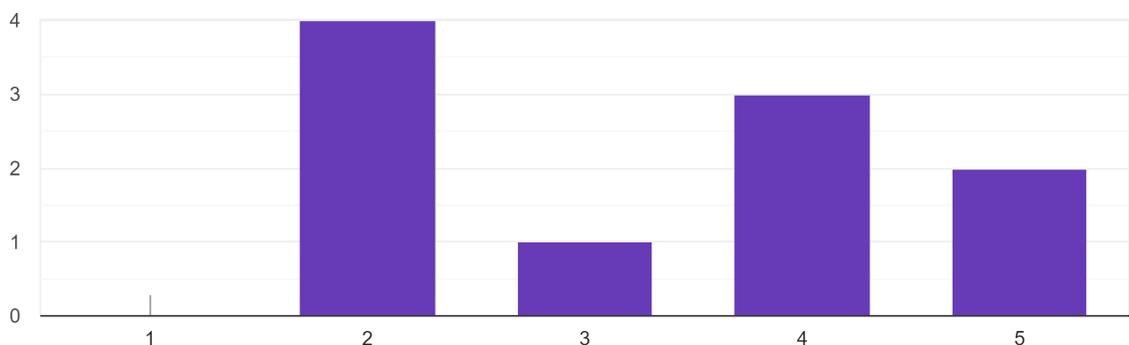


Figura 75: Porcentaje de jugadores que valoran el número de lugares recorridos en la historia, siendo el valor 1 equivalente a pocos lugares y 5 a muchos.

Por último en cuanto a las variables que componen el algoritmo y el sistema en el que se basa la clasificación de las narrativas, los personajes han obtenido unos valores diferentes en relación a la anterior prueba, ya que en esa encuesta se obtuvo que los personajes involucrados en esta narrativa eran más importantes que lo que reflejan los actuales resultados.

En la figura 76 se puede apreciar que la mayoría de los jugadores considera que los personajes de la historia no son muy relevantes, ya que el máximo se encuentra en 2 con 6 personas opinando que no tienen un nivel muy alto de relevancia, un valor entre bajo y promedio. Sin embargo, 3 personas han opinado que los personajes de la historia eran un poco importantes, teniendo el valor 4 un total de 3 votos. Por tanto se puede deducir que los personajes de esta historia no han dado la sensación de ser muy relevantes. Esto se puede deber a otros factores que no se han tenido en cuenta en el presente trabajo.

¿Crees que los personajes involucrados son importantes o dan información importante en la historia?

10 respuestas

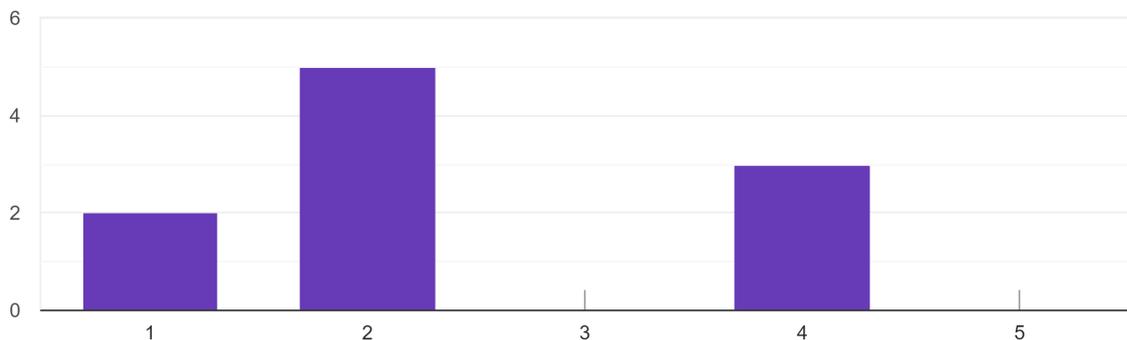


Figura 76: Porcentaje de jugadores que valoran la relevancia de los personajes involucrados en la historia, siendo el valor 1 poca relevancia y 5 una gran relevancia.

Por último, en cuanto a la comparación de por qué se han percibido las historias como principales o secundarias entre el motor computacional y los jugadores, representado en la figura 77, los resultados refuerzan la idea que se ha planteado en las anteriores historias.

La duración de la historia y la cantidad de personajes relevantes involucrados en la misma son los factores que más se han tenido en cuenta a la hora de decidir si una historia era relevante o no. En este caso la variable que más se ha tenido en cuenta es la cantidad de

personajes importantes, que tras relacionarla con los datos obtenidos en la anterior pregunta, se puede suponer que se debe a la baja importancia de los personajes involucrados en la trama.

La segunda variable más relevante a la hora de clasificar las historias según los jugadores ha sido la duración de la historia. Esto se ha debido a la localización de las diferentes partes de la historia y que era un poco complicada de seguir y completar.

Por último, aunque la diferencia con la duración de la historia sea de 1 punto, queda la cantidad de lugares visitados, que como se puede observar en el resto de historias se ha mantenido la variable menos relevante a la hora de identificar narrativas secundarias y principales.

El motor ha considerado que esta historia es SECUNDARIA, ¿a qué crees que se debe esta clasificación? (Puedes añadir otra si crees que afecta algo más)

10 respuestas

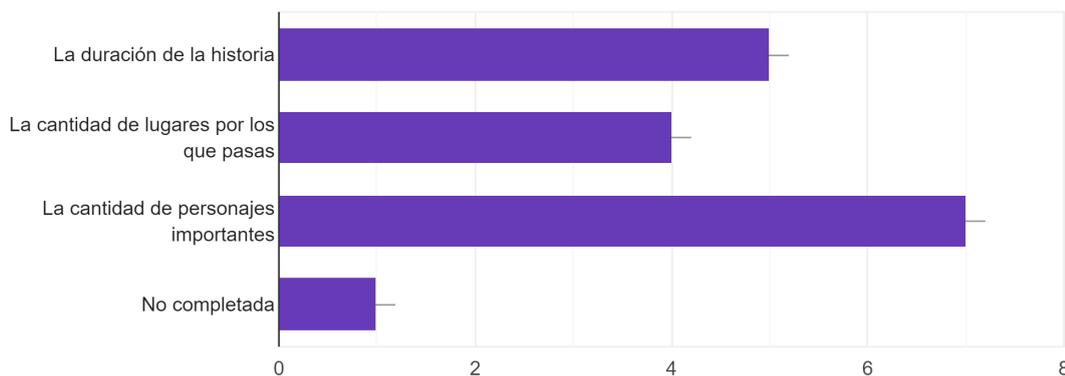


Figura 77: Porcentaje de jugadores que valoran las variables definidas en la hipótesis y algunas variables no estudiadas

8.2 Discusión de los resultados

Tras analizar los datos obtenidos de las encuestas realizadas por los jugadores en esta prueba final, se han obtenido resultados bastante diferentes a la anterior. Aunque hay algunos que es relevante destacar.

Para empezar, la importancia de los lugares visitados a la hora de clasificar una historia en principal o secundaria. Mientras que en la Historia 1, como se puede observar en

la figura 59, esta variable ha sido de las más relevantes para determinar que la historia era la principal, en el resto de historias se ha mantenido como una variable poco relevante a la hora de determinar su clasificación. Esto puede deberse a muchas causas, pero analizando los comentarios realizados por los jugadores se puede llegar a la conclusión de que los personajes tienen un impacto mucho mayor que esta otra variable y se suele tener en cuenta de forma prioritaria.

También ha influido la localización de inicio de la historia, que también ha sido un factor relevante para conseguir datos ya que no mucha gente ha podido jugar a todas las historias por no poder encontrarlas. Además de que muchos comentarios de los sujetos ha sido que la Historia 1 era principal porque era la primera que se encontraba en la sesión de juego, luego este factor se puede tener en cuenta en un proyecto posterior para colocar las historias dinámicamente de forma mucho más precisa para que consigan llamar la atención del jugador.

Además de esto, algunas personas han mencionado la importancia que han tenido algunos de los objetos que han aparecido en las diferentes historias y cómo podían prometer una historia mucho más extensa. Por ejemplo, la Historia 2 se ha descartado frente a la Historia 1 y la Historia 4, como pasó en la anterior prueba, porque los jugadores consideraban que lo que ocurría en ella no era muy importante ya que no involucraba nada más allá de rescatar a Julie, acto que se considera más un recado o un favor que una misión en sí que pueda iniciar una historia mucho más extensa y relevante.

Esto puede resultar de utilidad para futuros trabajos y proyectos, ya que si se tuviera un sistema que englobe más variables relacionadas con los objetos y la importancia de los diferentes elementos narrativos que componen una historia se podría conseguir una clasificación de narrativas mucho más precisa.

Por otro lado, en cuanto al funcionamiento del motor, que era el objetivo principal de esta prueba final, se ha podido comprobar que ha funcionado correctamente, ya que los jugadores han identificado las historias como principales y secundarias de igual manera que lo hizo el motor.

Además, los resultados de la encuesta refuerzan la teoría de que la importancia narrativa y diegética de los personajes es un valor muy importante a tener en cuenta para

clasificar las historias, ya que ha sido uno de los factores más importantes a la hora de clasificar las historias por los jugadores.

En cuanto a la situación de las historias en el juego y el momento en el que aparecen, habría que plantear un algoritmo que proporcionase historias y las pudiera colocar de manera que el jugador pueda interactuar con ellas.

Capítulo 9: Discusión

Tras haber analizado los resultados obtenidos en la última experimentación reflejada en el **Capítulo 8: Pruebas finales**, se comparan con los resultados obtenidos en la primera experimentación para comprobar si ha habido mejora al introducir en el sistema el motor de procesado de narrativa con su total funcionalidad. Para ello, en la primera parte de la discusión se relacionan estos resultados con los obtenidos en las primeras pruebas desarrolladas en el **Capítulo 6: Pruebas del prototipo**.

Una vez analizados dichos resultados, se relacionan con el desarrollo de todo el proyecto para tratar los aciertos de este y los puntos que podrían requerir de una futura mejora para ser más preciso o eficiente.

9.1 Discusión de los resultados tras la experimentación

Las primeras pruebas realizadas sirvieron para conocer la relevancia de las variables establecidas en la hipótesis, pudiendo así ajustar dicha hipótesis para lograr una distinción más precisa de las diferentes historias planteadas.

Este primer experimento refleja cómo han percibido los jugadores las diferentes variables propuestas, siendo estas la longitud de la historia y la cantidad de personajes involucrados. Si es cierto que la variable centrada en la cantidad de lugares recorridos no se analiza concretamente ya que se plantea como una variable que afecta a la percepción de la longitud de la historia por todo el recorrido por el que el jugador debe desplazarse.

De forma general, la longitud de la historia medida por los lugares y la cantidad de capítulos es captada, aunque se sustituyen los capítulos por escenas al ser más perceptibles por los jugadores. Por otro lado, el análisis de la longitud revela que podrían estar involucradas otras variables que no se tienen en cuenta en la hipótesis inicial, como la intervención de objetos claves en las tramas y la relevancia de los personajes involucrados en estas.

Debido a la importancia de esta relevancia y dado que la variable centrada en la cantidad de personajes no es percibida como relevante a la hora de clasificar las historias,

se sustituye el foco de dicha variable centrada posteriormente en estudiar la relevancia de los personajes.

Al reajustar la hipótesis y comprobar que se ajustan correctamente en las pruebas registradas en el **Capítulo 6: Pruebas del prototipo**, se realiza un segundo experimento para comprobar si la hipótesis redefinida se ajusta más que la anterior a la hora de clasificar las historias o, por el contrario, empeora dicha clasificación.

En este experimento el objetivo era comprobar si el motor había realizado bien la clasificación y si las historias estaban bien situadas dentro del juego después de que el motor proporcionase dichas historias.

Para ello se realizó una sección del cuestionario parecida a la del anterior para observar cuántas de esas historias se habían jugado, cuantas se habían percibido y cuánto tiempo había llevado completar el test. En este aspecto se han jugado bastantes historias, con la mayoría de los encuestados habiendo completado 3 o 4 historias en total, con porcentajes del 28,6% y el 21,4% respectivamente.

Sin embargo se han percibido muchas más historias, con 8 personas pensando que había por lo menos 5 historias. El tiempo en general que ha llevado completar este test ha sido variado, sin valores medios o concretos que sean remarcables para este trabajo.

En cuanto a los resultados obtenidos en las historias individuales, se puede observar que, como se predijo en los resultados del anterior experimento, la importancia de los personajes y su cantidad en relación con esta variable ha sido de gran importancia para determinar si las historias eran principales o secundarias, coincidiendo la gran mayoría de jugadores con la clasificación que había planteado el motor.

Por último, la situación de las historias dentro del juego se ha demostrado ser poco efectiva, en tanto que las historias secundarias se han jugado menos teniendo una de ellas un porcentaje del 50% de jugadores que han completado la misma.

Puesto que en este experimento el objetivo principal era comprobar el funcionamiento del motor en cuanto a clasificación de las narrativas, se puede concluir que es correcta, ya que la mayoría de jugadores han coincidido en los resultados que dio el

motor de la clasificación de personajes. Otro elemento que ha destacado en esta prueba ha sido la de la importancia de los objetos y de los lugares, algo que han mencionado algunos jugadores como parte relevante a la hora de diferenciar las narrativas por sus tipos.

Con ambos experimentos analizados, se procede a estudiar las diferencias existentes entre los dos experimentos y comprobar así si se ha logrado mejorar el sistema de clasificación de historias.

Comenzando por la longitud de las historias, se puede apreciar que no existen muchas diferencias. Si se analizan las diferencias entre las historias se puede apreciar que en ambos experimentos la primera historia, considerada como principal, tiende a considerarse como una historia de duración media según las figuras 34 y 56. Por el contrario, la segunda historia centrada en analizar dicha variable, es considerada como una historia corta según la figura 39 perteneciente a las primeras pruebas, pero tras las siguientes pruebas se percibe como algo más larga como se puede apreciar en la figura 62.

A pesar de que en la segunda experimentación estas cifras mejoran y se aproximan al objetivo a cumplir por dicha historia, a la hora de analizar la percepción de los jugadores vemos en la figura 65 que en la historia destaca la relevancia de los personajes por encima que la longitud de esta a la hora de clasificar dicha historia como secundaria, por lo que podemos afirmar que el diseño de la historia no cumple con los objetivos propuestos a pesar de mejorar su percepción en el segundo experimento.

A pesar de ello, su valor sigue siendo alto comparado con el del resto de historias, por lo que la longitud de la historia es una variable a tener en cuenta a la hora de clasificarla. Como se ha podido apreciar, el conflicto a la hora de analizar dicha historia es la intervención del resto de elementos que cambian percepción de la longitud, como ha sido el caso de los lugares recorridos o la relevancia de los personajes involucrados en esta, que son separados en la segunda experimentación, pero no se tienen en cuenta todas las variables, como los objetos clave, por lo que esta variable debe ser más precisa y, por tanto, analizar de forma separada de cada uno de los elementos que influyen a la hora de ser percibida por los jugadores.

En cuanto al análisis de la cantidad de lugares recorridos, en el primer experimento son tratados de forma implícita como parte de la longitud de la historia, por lo que la primera

prueba no analiza esta variable de una forma más concreta. Sin embargo, en la segunda experimentación si son tratados individualmente, por lo que podemos apreciar la verdadera relevancia de esta variable en la clasificación.

En la historia principal, la cantidad de lugares percibidos según la figura 57 es media-alta, mientras que la historia 4 centrada en estudiar esta variable recorre una cantidad media de lugares como se puede apreciar en la figura 75. El resto de historias mantienen una cantidad media baja, por lo que la historia 4 cumple con su objetivo a primera vista.

Si analizamos la información de la figura 77, la mayoría de los jugadores determina que la clasificación de la historia se debe más a la relevancia de sus personajes que a la cantidad de lugares recorridos, por lo que esta tampoco cumple completamente con los objetivos establecidos para ella. Aún así, podemos afirmar que la influencia de la cantidad de lugares, a pesar de ser menos determinante que otras variables, sigue teniendo una gran relevancia a la hora de clasificar las historias.

Por último, la variable centrada en estudiar la relevancia de los personajes involucrados en la trama se analiza de dos formas diferentes según la fase de experimentación en la que es estudiada.

En la primera experimentación, se analiza la cantidad de personajes involucrados en el análisis de la tercera historia. Esta cuenta con una gran cantidad de personajes, aunque con una duración corta que en combinación con la cantidad de personajes la convierten a ojos del jugador en una historia secundaria, no como en el caso de la historia principal que mantiene ambas cifras en una cantidad media como se puede ver en la figura 45.

Por otro lado, en la segunda experimentación se analiza la relevancia de los personajes en vez de su cantidad. La historia principal cuenta una relevancia de sus personajes elevada como se puede apreciar en la figura 58, siendo además la más destacada entre las variables según la valoración de los jugadores en la figura 59.

Por otro lado, la tercera historia cuenta con una relevancia media según la figura 70, con cifras más bajas que la historia principal pero manteniendo como variable destacada la relevancia de personajes frente al resto de opciones.

Tras este análisis, la relevancia de los personajes no solo influye más a la hora de clasificar historias, si no que es más precisa que la cantidad de personajes. Por otro lado, la relevancia de estos es un factor más perceptible al desarrollar las tramas de las historias y, por ello, es la variable más destacada de todas las historias, por lo que esta variable afecta en mayor medida a la hora de clasificar la historia que el resto de variables propuestas.

Tras analizar todos estos aspectos, se determina que las pruebas han logrado dar con variables relevantes a la hora de distinguir los tipos de historia. Sí es cierto que solo una de las historias cumple verdaderamente con la demostración total de la variable que analiza, siendo esta la calidad de los personajes de una historia. Esta variable es la más destacada a la hora de diferenciar historia y, aunque el resto de variables también son relevantes, esta alcanza un nivel muy alto en algunas ocasiones y no permite estudiar correctamente la influencia del resto de variables.

Un planteamiento para poder enfocar este conflicto sería analizar la calidad narrativa de los personajes de forma independiente a su relevancia diegética y la influencia que tienen en el mundo del juego. También sería interesante analizar ambas relevancias junto a la cantidad de personajes, analizando así las tres variables que encapsulan a los personajes.

No por ello se descarta el resto de variables, la cantidad de lugares recorridos también es un elemento relevante a la hora de interpretar historias y que, gracias a la segunda experimentación, se refleja mejor la importancia que tiene realmente esta variable separándose de cómo afecta a la longitud de la trama , como ocurre en el primer experimento.

Esta longitud sin embargo, es una variable que a pesar de ser relevante, no es percibida de la misma manera que el resto de variables, por lo que los experimentos no logran determinar una relevancia precisa de esta variable.

9.2 Discusión general del proyecto

Tras haber realizado el análisis de los datos obtenidos en las dos pruebas realizadas y habiendo comparado los resultados de ambas, se puede hacer un análisis del propio proyecto y los objetivos que se han cumplido, ya que ya se cuenta con información suficiente como para valorarlo y sacar conclusiones.

Puesto que el objetivo del proyecto era desarrollar una primera versión de un motor computacional que fuera capaz de clasificar las diferentes narrativas que se le proporcionasen, se puede concluir que el objetivo se ha cumplido. Actualmente, el motor es capaz de diferenciar entre dos historias y determinar cuál es principal y cuál secundaria según los parámetros proporcionados y que se plantearon en la hipótesis de partida y se modificaron más tarde tras los resultados de la primera prueba.

Además, se han obtenido datos que pueden ser relevantes para futuras ampliaciones y que no estaban previstos, como los objetos o los lugares y sus respectivas importancias.

Sin embargo, estos datos pueden variar de un videojuego a otro, como se mencionaba en el **Capítulo 2.1.2: Narrativa en los videojuegos**, ya que cada género tiene una forma distinta de contar una historia. Mientras los juegos de tipo *sandbox* no tienen una narrativa definida y se guían más por la narrativa emergente, los juegos más narrativos como son los juegos de rol o las aventuras gráficas *Point & Click* siguen una narrativa preestablecida que puede ser más o menos compleja pero que no deja mucha más experimentación a los jugadores.

Por tanto, tras los resultados obtenidos sería interesante aplicar este motor a diferentes géneros y realizar un estudio de qué variables son las que afectan a cada género, con tal de hacerlo más preciso y abierto, ya que en este trabajo sólo se ha enfocado en juegos de un estilo más clásico de rol.

Esto resultaría en una mejora muy interesante para el motor y que ampliaría sus posibilidades a muchos más juegos, permitiendo crear narrativas en todo tipo de experiencias y géneros.

Otra mejora que resultaría muy interesante sería la de aplicar las importancias de objetos y lugares, como ya se ha mencionado previamente, que mejoraría la precisión con que el motor calificaría las narrativas, proporcionando una mejor experiencia a los jugadores y pudiendo afinar todavía más los resultados.

Añadido a esto, se podría estudiar la relación que existe entre las variables que se han aplicado en esta implementación del motor, como son la longitud de las historias y la importancia de los personajes. Esto es debido a que muchas personas han percibido

algunas historias más largas porque involucran personajes con un mayor peso narrativo y diegético. Por tanto, conseguir conocer la relación que existe entre estas variables podría ser de mucha utilidad para volver más preciso el valor de longitud de una narrativa.

A esto también se suma, el interés de la importancia de los objetos. Muchas personas han mencionado que los objetos que eran más importantes daban más sensación de relevancia a una historia. Por ejemplo, en la historia 4 se mencionó la existencia del objeto que menciona el padre de Kanda a su hijo y que eso podría significar que la narrativa iba a tener un peso más importante. Esto se puede unir con algo que se menciona en técnicas narrativas y en teoría de escritura: los McGuffin.

Este término fue acuñado por Alfred Hitchcock y es mencionado en el libro *El cine según Hitchcock* (Truffaut et al., 1984). Estos son objetos, situaciones o elementos que hacen avanzar la trama pero que no son relevantes por sí solos y que si se cambiaran por otra cosa completamente distinta, pero con la misma funcionalidad, seguirían funcionando para conducir y mover la trama. Por ejemplo, la trifuerza en *The Legend of Zelda* (Miyamoto et al., 1986) actuaría a modo de McGuffin, ya que se puede sustituir por cualquier otro objeto y la historia seguiría funcionando, y al mismo tiempo funciona como elemento conductor de la misma.

Esto explicaría la importancia que los jugadores le han dado a los diferentes objetos que han aparecido en las distintas narrativas. Por tanto, aplicar al motor una manera de diferenciar los diferentes objetos que componen una trama y darles relevancia para que así puedan funcionar como un elemento más a tener en cuenta en la clasificación de historias haría de está algo mucho más preciso.

Destacar también la importancia de los lugares, ya que esto también puede resultar relevante y complementar a la variable de cantidad de lugares que se han visitado en la historia. Además, esta importancia se podría relacionar con la de los objetos, ya que si un lugar contiene un objeto de alta relevancia, este lugar tendría una importancia mayor que otros.

Relacionado también con los lugares, ha surgido un sesgo, que no se tuvo en cuenta en un primer momento, a raíz de las pruebas experimentales. Éste es la situación de las historias respecto al personaje y al lugar de inicio de las mismas.

Muchas personas han identificado la narrativa principal como tal porque era la primera que se encontraban nada más empezar a jugar, por lo que asumían directamente que esta era la principal. Por tanto la cercanía a los personajes no jugables que inician las historias es algo relevante también a la hora de discernir si una narrativa es principal o secundaria.

Por tanto, tener en cuenta este dato y aplicarlo al motor computacional podría ser también de utilidad para mejorar la clasificación de las historias. Sin embargo, en estas pruebas eso no se tuvo en cuenta, por lo que es posible que esto haya influido en los resultados obtenidos en las pruebas.

Para poder realizar estos cambios y comprobar de qué manera afectan a la trama, a la relevancia de la misma y a su clasificación sería necesario realizar más investigaciones y pruebas en entornos controlados y de esa manera ver cómo se comportan estos datos y de qué manera interactúan con los demás planteados en este trabajo. Además se podría plantear diferentes niveles dentro de la clasificación, teniendo incluso narrativas terciarias o distintos niveles dentro de las narrativas principales.

Capítulo 10: Conclusiones y trabajo futuro

La investigación realizada sobre narrativa en videojuegos ha permitido el desarrollo de un motor de identificación y procesado de historias, que es capaz de categorizar narrativas en principales o secundarias. Se han estudiado varios componentes narrativos para determinar su relevancia en la categorización, escogiendo un número limitado para poder acotar el experimento a las variables que consideramos más destacables: número de escenas, número de lugares visitados e importancia narrativa de los personajes involucrados.

El motor es fácilmente escalable para tener en cuenta nuevos factores narrativos, mientras estos sean cuantificables y se puedan calcular con los datos existentes en la estructura actual. Aún así, el modelo propuesto es simple y su ampliación puede aportar mayor potencial al motor.

En futuros trabajos, se pueden aprovechar los datos recabados para mejorar los sistemas de generación procedural que, hasta la fecha, no se han orientado a introducir tramas de menor relevancia para acompañar o asistir a la principal.

10.1 Conclusiones del proyecto

A la hora de investigar sobre la narrativa computacional existe una vasta documentación sobre la generación de historias de una trama principal única, es decir, situaciones narrativas en las que solo existe una historia principal por lo que no tendría cabida la definición de historias secundarias. Este hecho contrasta con la realidad en el mundo de los videojuegos, donde la existencia de historias secundarias es habitual.

Ambos factores unidos hacen que la presente investigación tenga una especial relevancia, ya que por un lado hay poca información sobre la generación procedural de narrativas secundarias, y por otro lado hay poca experimentación tratando de generar tramas secundarias en videojuegos.

Pasando a otra de las conclusiones extraídas, tras la primera fase del experimento que corresponde con el **Capítulo 6: Pruebas del prototipo**, se pudo observar que las variables escogidas para la hipótesis inicial, número de personajes, número de capítulos y

lugares visitados, no son las más significativas para el correcto funcionamiento de un sistema de categorización automatizado.

Por otro lado, a partir del análisis de los resultados obtenidos en la segunda fase, se ha podido deducir que el número de lugares visitados sí es una variable relevante a la hora de la categorización. En contraposición, se han descartado los otros dos factores analizados en la primera fase, pudiendo ser intercambiados por el número de escenas y la importancia narrativa de los personajes, que sí han resultado ser cruciales a la hora de clasificar una historia en principal o secundaria.

Por su parte, y tras el análisis de las encuestas, la relevancia de los personajes ha sido destacada por los sujetos de estudio, que también resaltaron la importancia narrativa de los objetos. Asimismo, también se han localizado otros factores como la facilidad para empezar cada historia o la posibilidad de continuación al llegar al final de la trama. Todos estos factores extrapolados de las encuestas pueden ser nuevas líneas de investigación futuras y se desarrollarán más adelante en el **Capítulo 10.2: Trabajo futuro**.

En cuanto al desarrollo del motor, el sistema construido se ha podido comprobar que es capaz de estructurar las historias en cualquier proyecto con las mismas tecnologías en el que se integre. Debido a su versatilidad, el uso de las variables para realizar los cálculos del procesado de historias es sencillo y directo, por lo que la inclusión de futuros componentes narrativos será dependiente de la complejidad del propio factor en mayor medida que de la implementación en el motor.

En otra línea, el uso de Unity para agilizar la implementación del prototipo con el que se han realizado las pruebas ha permitido que se dedicase más tiempo a la investigación y el desarrollo del motor. Aprovechando el modelo que construye el motor, el prototipo ha sido rápido de desarrollar para poder realizar la primera prueba lo antes posible, y fácil de adaptar a los cambios que se produjeron para la segunda prueba.

Como conclusión final, se podría destacar que se precisa de más investigación, tanto de las variables estudiadas en este proyecto como de las que han surgido durante el análisis de resultados, para mejorar la precisión de la categorización actual. En esta misma línea, se deduce que son necesarios más componentes narrativos de los actualmente aplicados en el sistema para poder profundizar en el procesado.

Asimismo, con esta investigación se ha pretendido marcar unos primeros pasos en el estudio sobre narrativa secundaria de videojuegos que puedan ser ampliados en futuras investigaciones, potenciando la generación de narrativa y dando más posibilidades a su aplicación en los videojuegos.

10.2 Trabajo futuro

A la hora de estudiar los componentes y la estructura de la narrativa, se identificaron una serie de variables que podrían ayudar a precisar la identificación de las historias. Debido a las limitaciones de tiempo del proyecto, el equipo se vio obligado a escoger las consideradas como más relevantes para acotar el estudio y poder hacer un análisis más exhaustivo.

La primera variable que se identificó como potencialmente relevante es la importancia narrativa asociada a objetos clave de las historias. Se ha comprobado en la experimentación que los jugadores han considerado más relevantes las historias que incluían un objeto de gran valor o poder. Algunos comentan que, debido a estos objetos, la historia podría ser la principal.

Esta variable se podría estudiar de forma similar a la de importancia narrativa de los personajes, asignando un valor numérico por el que el motor pueda calcular el impacto que ejerce sobre la historia. Sería necesario realizar más pruebas con casos específicos para comprobar que es una variable determinante.

Otro de los factores que se consideran interesantes de estudiar en el futuro es la introducción de la historia al jugador, es decir, lo fácil que lo tiene el jugador para iniciar dicha historia. Esta variable se ha identificado tras el estudio del primer experimento, pues varios encuestados comentan que la historia 1 es la principal pues el personaje que la inicia se encuentra justo al salir de la casa del protagonista, mientras que otras historias debían ser secundarias al obligar al jugador a explorar o salirse de los caminos principales para comenzarlas.

Esta variable representa ciertas complicaciones de cálculo en el estado actual del motor. Actualmente, la estructura de datos de las historias contiene los personajes que

existen en cada lugar, pero la posición real en el juego, las coordenadas, son ajenas al motor. Por lo tanto, aunque se calculase la distancia en lugares que existe entre la posición inicial del jugador y el inicio de cada historia, si varios inicios se sitúan en un mismo lugar, no se obtiene el dato correcto para realizar el cálculo. Como se puede apreciar en la figura 78, la distancia en el juego entre la posición inicial del jugador y los personajes que inician las historias 1, 3 y 4 es muy distinta. Sin embargo, el motor entiende que los 3 personajes están a la misma distancia.

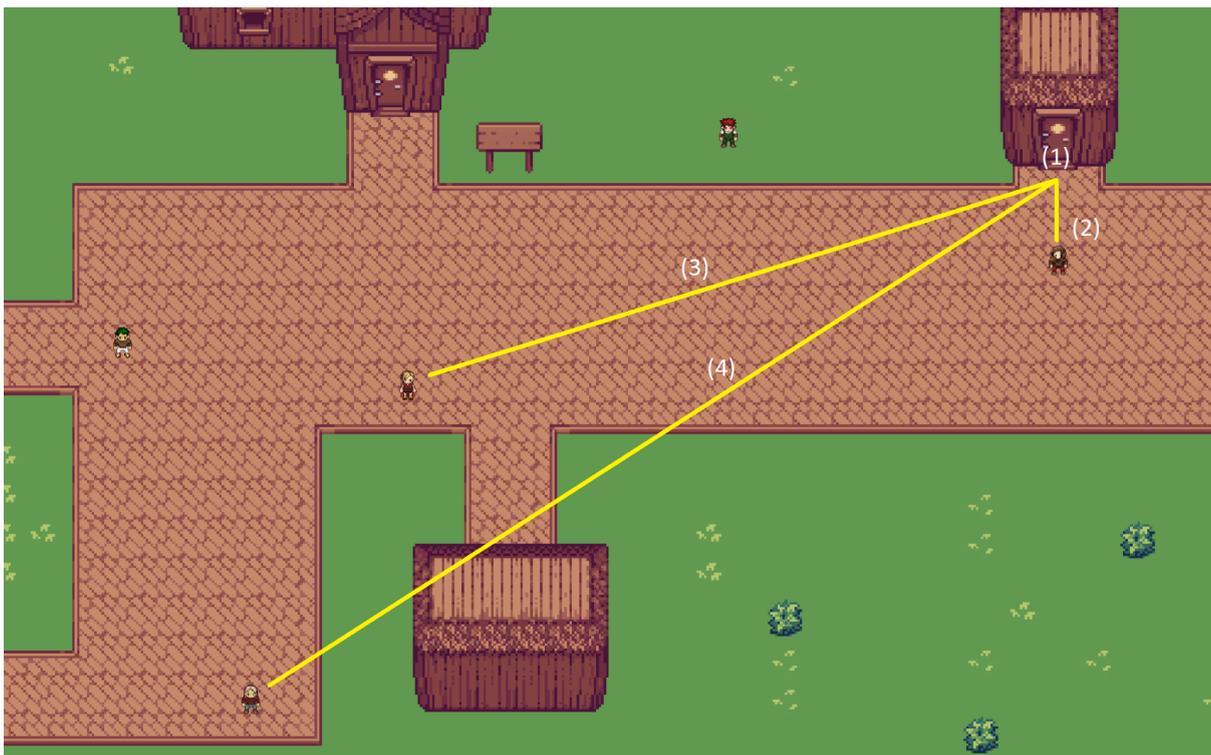


Figura 78: Imagen del prototipo de juego del lugar “Fyrst”, en el cual (1) es la posición del jugador al salir de su casa, (2) es la distancia desde (1) hasta el personaje de la historia 1, (3) es la distancia desde (1) hasta el personaje de la historia 3 y (4) es la distancia desde (1) hasta el personaje de la historia 4.

Aún así, dados los resultados de los experimentos, este factor se considera muy relevante para la clasificación de historias y no se descarta realizar modificaciones al modelo actual para poder estudiar el impacto real que supone sobre la percepción de los jugadores.

Para finalizar con las propuestas de factores narrativos que se han detectado durante el experimento, varios de los encuestados comentaron que las historias 1 y 4 podrían ser consideradas principales al poseer finales abiertos, es decir, la trama podría continuar en el futuro aunque no estuviese implementado en el prototipo. El análisis de esta

variable es de nuevo complejo, al no existir un indicativo claro de que la historia podría continuar y existen muchas formas de dejar tramas abiertas a continuación.

Además del análisis de variables, se ha considerado la posibilidad de establecer en el motor un “umbral” de historias principales, por el que varias historias que tuviesen una valoración superior a una cantidad fuesen categorizadas como tal, dando lugar a más de una historia principal en el juego, ya sea de forma paralela o en situaciones diferentes.

Aunque incluir un valor en el motor que sirva de diferenciador entre narrativas principales y secundarias puede parecer trivial, hay que tener en cuenta la naturaleza de los cálculos que realiza el motor para la clasificación de historias. La puntuación de cada historia se obtiene estudiando cada variable en comparación con los valores acumulados de todas las historias, por lo que tener una gran cantidad de historias hará que los valores sean muy pequeños y tener muy pocas historias tendrá el resultado contrario. Además, si las historias son muy similares en cuanto a métricas, las puntuaciones podrían ser muy similares y que todas, o ninguna, fuese categorizada como principal.

Una posible solución a la asignación del valor de corte sería calcular el umbral de forma dinámica en base a los datos de las historias en cada carga, al igual que el cálculo se ajusta a los datos actualmente al calcular en base a los valores acumulados. De igual manera, se puede limitar el número de principales estudiando el número de historias y sus puntuaciones, asegurando que sólo las que son relevantes son categorizadas como tal.

Si se profundizase en la relación entre historias principales y secundarias, se podría implementar un sistema que, a partir de la situación del jugador en la historia principal, se introdujesen las secundarias en el juego de la forma más natural y orgánica posible, con el posible objetivo de que las secundarias asistan a la principal. Esto podría traducirse en redirigir al jugador a su siguiente escena en la principal, o facilitar el progreso si, por ejemplo, se identifica que el jugador tiene problemas para superar los desafíos (combate, puzzles, etc.) de la principal.

Capítulo 11: Contribución individual

11.1 Alejandro Marín Pérez

Al principio Alejandro estuvo junto a todos los compañeros buscando e investigando acerca de narrativa computacional y creatividad computacional. Adicionalmente también estuvo investigando por otro lado para profundizar más acerca de narrativa, leyendo manuales y libros de escritores en los que explicaban su proceso creativo por si eso servía de utilidad durante el desarrollo del trabajo.

Puesto que una de las primeras preocupaciones que hubo al comienzo fue acerca de la representación computacional de los elementos narrativos, también se estuvo investigando cómo se representa esto en videojuegos, buscando diferentes charlas de la GDC (Games Developers Conference, Conferencia de Desarrolladores de Juegos) que pudieran ser de utilidad, sobre todo las que hablaban de la narrativa en videojuegos como *Fallout* o *Grim Fandango*.

Conforme se iban encontrando artículos o charlas que fuesen útiles se compartieron con los compañeros para que todos partieran de la misma base a la hora de entender el trabajo y cómo enfocarlo.

También fueron muy útiles algunos vídeos de tipo ensayo que hablaban de narrativa en videojuegos, analizando cómo enfoca cada juego su forma de contar una historia y de qué manera afecta eso al jugador y a cómo percibe este la narrativa que se le presenta. Estos vídeos están sobre todo sacados de canales como *Game Maker's Toolkit*, *Design Doc* o *noclip* entre otros. Aunque fueron de utilidad para entender el funcionamiento de la narrativa en videojuegos e intentar buscar patrones de escritura que como investigación per sé en cuanto a narrativa computacional y escritura.

Tras realizar la investigación en este aspecto era necesario plantear la estructura interna que iba a tener el motor computacional para representar las historias y poder procesarlas. En este aspecto se plantearon varios problemas, acerca de qué elementos eran los más necesarios representar y de qué manera.

Para ello se acudió a los manuales de escritura que habían sido consultados y a los diferentes libros acerca del proceso de escritura que habían seguido algunos escritores y cuáles recomendaban, como los que planteaban varios escritores en el podcast llamado *Writing Excuses*. De estas referencias se encontraron unos programas de escritura, softwares que facilitan la escritura de novelas proporcionando una serie de herramientas al autor para que no tenga que tener todo separado en diferentes documentos. Estos programas eran *Bibisco* y *Scrivener*. Usándolos como referencia, se planteó la posible estructuración de las narrativas en Historias - Capítulos - Escenas, además de extraer de ellos también el tener en cuenta a los personajes y los lugares, ya que cuentan con una funcionalidad para describirlos y tener fácil acceso a ellos y a sus diferentes características.

Tras plantear la estructuración de las historias y de que se plantease la hipótesis de partida, se pasó al desarrollo del prototipo para probar que esta funcionase correctamente. En esta parte se encargó de desarrollar el sistema de diálogos y de plantear la estructura de los JSONs de diálogos en nodos de diálogo que contenían un texto, el personaje que lo dice y las diferentes opciones, así como un sistema para detectar cuándo se ha terminado la conversación y si se ha hecho correctamente.

Para ello desarrolló un sistema sencillo que mostraba el texto que decían los personajes y que se pudiera pasar y cambiar entre diferentes textos y comentarios de personajes dentro del juego, para luego implementar el inicio de estos diálogos con los personajes.

Además de esto, también incluyó un sistema de mostrado de las opciones de los diálogos para que cupieran todas en pantalla y no se saturase al jugador con demasiadas opciones. Este sistema muestra tan sólo tres opciones pero puede llegar a mostrar muchas más sin tener límite, ya que si un nodo de diálogo tiene más de tres opciones divide todas en grupos de tres opciones y las va mostrando a medida que el jugador navega por ellas, pudiendo volver a la primera opción desde la última y viceversa.

De manera que todo funcionase correctamente también implementó el sistema de cargado de los diálogos de JSON a Unity y posteriormente de JSON al motor computacional, para que fuese este el que proporcionase los diálogos al juego. Para que esto funcionara correctamente hubo que modificar un poco la implementación hecha en Unity, ya que la forma de interpretar y cargar los JSON cambiaba un poco entre el motor y Unity.

Además de eso, contribuyó también en la implementación del sistema de gestión de escenas del motor, ya que dependían un poco de los diálogos y era necesario saber cuándo terminaba uno para pasar a la siguiente escena. Esto llevó un par de modificaciones en los diálogos, cambiando el número que indicaba el final de diálogo para tener dos, uno que indicara un final de diálogo satisfactorio y otro que indicara un final no satisfactorio. Esto haría avanzar la historia o no en función del resultado obtenido en la conversación. También fue necesario fusionarlo con el resto de sistemas del motor para que todo funcionase correctamente.

Paralelamente se ocupó de la redacción de las diferentes historias, atendiendo a las restricciones que habíamos planteado para medir las variables de la hipótesis de partida y que tuvieran sentido como historias independientes además de aportar los datos necesarios para avanzar en el trabajo.

Tras redactar un primer borrador de las narrativas y comprobar que estaban bien escritas para los fines que se buscaba en el trabajo, pasó al desarrollo y la escritura de los diálogos en JSON, revisando que el formato fuera el correcto y que no produjese ningún error ninguno de los archivos creados. Este proceso fue bastante más largo de lo esperado debido a las revisiones que hubo que hacer de erratas, conexiones entre nodos y correcto funcionamiento del prototipo.

Tras tener todo el prototipo listo, participó también en la distribución y búsqueda de personas que pudieran rellenar las encuestas y de esta manera obtener los datos necesarios para hacer avanzar el proyecto.

Una vez obtenidos los datos, tomó parte junto a sus compañeros en el análisis de estos datos y en la obtención de las conclusiones, como fueron la importancia de los personajes y de los objetos a la hora de determinar la importancia de las narrativas. Para implementar estos cambios planteó la introducción de las diferentes escalas de importancia de los personajes y la forma de relacionarlas, usando una media ponderada para la que se podría realizar un estudio con tal de obtener los mejores valores de dicha ponderación y conseguir un valor de relevancia de personaje mucho más preciso para usarlo en la clasificación de las narrativas.

Finalmente participó junto a sus compañeros en la redacción y corrección de la memoria, añadiendo referencias y bibliografía, además de revisar que el formato fuese el mismo a lo largo de la memoria, aunque esto fue un trabajo de todos los integrantes del grupo.

11.2 Carlos María González Martínez

En un principio, Carlos María González se dedicó junto a sus compañeros a pensar en la ampliación de la historia del prototipo, con tal de tener una con la que crear el resto de historias. Tuvo la idea de hacer a los héroes y villanos como trabajadores asalariados o por prestigio, que evolucionó en la historia de fondo que se usa en la versión final.

Paralelamente, empezaron con el desarrollo del prototipo de juego en Unity. Carlos María González se encargó de la creación inicial del proyecto, y de desarrollar varios sistemas para el funcionamiento correcto del mismo. En específico, empezó realizando modificaciones al sistema de interacciones. Originalmente otro de sus compañeros hizo una versión tentativa que utilizaba proximidad entre el personaje jugador y el objeto interactuable para detectar y saber si permitir la interacción o no, pero esto, además de integrarlo con el primer estado del sistema de movimiento, fue cambiado por Carlos María González para que utilizara el sistema de colisiones del motor físico de Unity. Con esto, se aseguraron de que las interacciones no tuvieran conflictos por varios personajes en la misma zona, y requerían de un contacto directo entre ambos objetos, mucho más fácil de detectar que por proximidad.

Alejandro Ruiz y Carlos María González se encargaron del sistema de misiones, crearon dos componentes de clase NPC (objetos interactuables): Los QuestStarterNPC inician la misión y los diálogos que comunican el objetivo, y los QuestFinisherNPC, los cuales determinaban si la misión se había completado con éxito y daba la misión por concluida. En este estado todavía no empleaba el uso de misiones del motor. La funcionalidad de dicho sistema se ha ido modificando junto al desarrollo del motor y el avance del proyecto: Las misiones se componían de un número indeterminado de escenas, por lo que se extendió el rol de QuestFinisherNPC, para que la clase también pudiera tratar con escenas intermedias e indicar que se podía pasar a la siguiente.

La tarea de desarrollo de sistemas que se le asignó exclusivamente a Carlos María González fue el desarrollo del sistema de lugares. Originalmente creó lugares con programación simple, empleando dos coordenadas (esquinas superior izquierda y la inferior derecha) para determinar su área y localización, para así poder detectar la presencia del jugador en la zona. Una vez se creó el motor, expandió la funcionalidad del sistema: Cuando detectaba al jugador en el área, procedería a comunicarse con el sistema de misiones para

la instanciación de las correspondientes a historias que empezaban en ese lugar. En posteriores versiones también cambió a pedir solo una cantidad reducida de misiones cada vez que se entraba, así como la solicitud de diálogos genéricos para los personajes presentes en la primera visita. Además, añadió una lista que contiene todos los personajes presentes (incluyendo objetos inanimados).

Hay dos funcionalidades más del prototipo que él programó que utilizan los tres sistemas recién explicados. Hay personajes que tienen papeles en varios lugares diferentes, así que tuvo que implementar en sus clases un sistema que comprueba, una vez está fuera de cama del jugador, si se encuentra en el lugar en el que debería estar para el progreso de la misión. En caso de no estarlo, se teletransportará al lugar correspondiente, y utilizará una de las coordenadas válidas, guardadas en el lugar. para colocarse. La segunda funcionalidad tenía que ver con ciertas escenas en las que el personaje protagonista tenía pensamientos que no se correspondían a ningún otro personaje u objeto. Por ello implementó un sistema para que estos diálogos se iniciaran instantáneamente al momento en el que el personaje entrase en un lugar y se cumplieran las condiciones para su activación, para proseguir con las misiones como una escena más.

Durante el desarrollo del motor, Carlos María González se encargó de crear el sistema de carga y lectura de ficheros en formato JSON, los cuales contenían toda la información de la historia del prototipo. En un principio lo desarrolló con la funcionalidad ya creada en .NET Core 5 para el tratamiento de JSON, pero por incompatibilidad actual con Unity, tuvo que reescribir la lectura de ficheros para .NET Core 3.1. En un principio cargaría toda la historia, personajes y lugares, y por último leería archivos de diálogos por demanda del sistema de diálogos. Una vez la carga de historias, capítulos, y escenas estaba terminada, también se encargó de programar su enlazado, es decir, asignar escenas a sus capítulos correspondientes, y asignar capítulos a la historia que componen.

También fue responsable de la compilación del motor en una librería dinámica, y su posterior enlazado a Unity. Empleando la clase principal del motor como frente, creó un sistema de comunicaciones entre el motor y el prototipo. Hizo que el prototipo inicializara el funcionamiento del motor, y a partir de ahí, se le llamara en distintos puntos del juego. Cuando el jugador entraba en un lugar, el sistema de misiones solicitaba una lista de capítulos que empezaban allí, y con ellas fabricaba misiones del prototipo. Estas misiones tendrían en cuenta el personaje que las empezaba, y si no disponía de un componente NPC

adecuado, los crear para asignarles la misión. También hizo que el sistema de diálogos llamase al motor en los momentos adecuados, tales como la carga y creación de misiones, y la carga de diálogos genéricos. Algunas partes de esta implementación serían cambiadas para la segunda versión del prototipo, debido al cambio de la hipótesis.

Respecto a la nueva hipótesis, Carlos María González se encargó de implementar su correcta carga en el motor en base a los JSON de las historias y los personajes. Se encargó de programar el cálculo de importancia de las historias de acuerdo a la fórmula, y su aplicación para ordenar las historias por su prioridad, así como determinar la más importante y denominarla como principal.

A lo largo de todo el proyecto, ha estado depurando y arreglando errores de programación tanto propios como ajenos, incluso aquellos difíciles de detectar, ya que su entorno de trabajo (Rider) era más útil para dicha tarea. Y por último, también se dedicó a limpiar el código para tener un formato más ordenado, consistente, y legible.

11.3 Carlos Jiménez Álvarez

Al comienzo del proyecto, al igual que el resto del equipo, Carlos dedicó las primeras semanas a recabar información y bibliografía. Al provenir de Ingeniería de Software, le faltaban muchos conocimientos básicos de narrativa que poseían sus compañeros de Videojuegos, por lo que se informó mediante libros, artículos y conferencias que recomendaron tanto el tutor como los compañeros. También buscó ejemplos de videojuegos que sirvieran para ejemplificar los distintos conceptos que se iban a estudiar.

Una vez reunida la información necesaria para formular una hipótesis, se empezó con el desarrollo del prototipo de juego en Unity y la primera versión del motor, que se limitaría a cargar y estructurar todos los datos y ofrecer al prototipo funcionalidades para poder leer las historias y sus componentes. Al ser un juego en 2D, el equipo buscó algunos assets (spritesheets) que sirviesen para construir el mapa y diferenciar a los personajes, decidiendo usar unos packs adquiridos a través de Humble Bundle y un generador personalizable online de spritesheets de personajes, Universal LPC Spritesheet Character Generator¹⁶. Carlos creó los spritesheets de personajes y escogió los que se usarían para el mapa.

Se encargó de implementar el sistema de movimiento del personaje principal en Unity mediante el uso del nuevo Input System, el cual permite crear mapas de controles, asociarlos a un gameObject que actuaría de controlador y conectar las distintas acciones a métodos definidos en el script de PlayerMovement.

Para la creación del mapa de juego, creó Tile Palettes en Unity que permiten organizar los Tiles de forma sencilla, lo que agiliza el proceso de “pintado”. Para ello, también tuvo que ajustar cada spritesheet para que los Tiles fuesen uniformes y tuviesen el tamaño y calidad que se quería. El mapa lo montó usando dichos Tile Palettes, añadiendo a la escena de Unity un gameObject con componente Grid rectangular, que permitiese pintar el mapa por casillas rectangulares (los Tiles), y tres hijos con componentes Tilemap para diferenciar la capa del Background (el suelo), el Foreground (los caminos o elementos decorativos como hierba o flores) y los obstáculos (mobiliario, carteles).

¹⁶ <https://sanderfrenken.github.io/Universal-LPC-Spritesheet-Character-Generator/>

Como se ha comentado anteriormente, la creación de los spritesheets de los personajes se ha hecho mediante la página Universal LPC Spritesheet Character Generator, resultando en 38 spritesheets de personajes diferentes. De estos spritesheets, finalmente se decidió usar solo el sprite básico de cada personaje al no introducir ningún tipo de animación en el prototipo que pudiese retrasar el desarrollo del proyecto.

Para poder crear entornos interiores en el prototipo de juego, implementó un sistema de uso de puertas por el que, al interactuar con ellas, se teletransportase el personaje principal a la posición asignada a dicha puerta. Esto permitió crear las casas como escenarios aislados del mapa principal y de esta forma evitar realizar cargas innecesarias de escenas de Unity.

También implementó un sistema de movimiento aleatorio para los personajes no jugables, es decir, que estos personajes se moviesen en direcciones aleatorias durante distancias aleatorias dentro de un rango de distancia desde el punto de origen del personaje, a intervalos regulares definidos y a una cierta velocidad. Esta funcionalidad se podía activar o desactivar a cada personaje no jugable individual, de forma que unos fuesen deambulando mientras otros estuviesen estáticos en su posición original. Al interactuar con los personajes que estuviesen deambulando, estos se quedarían quietos hasta el fin de la interacción, y si colisionasen con algún obstáculo u otro personaje no jugable la dirección de movimiento se invertiría hasta que se cubriese la distancia designada de movimiento. Este sistema, aunque implementado en el prototipo, no se llegó a usar al no crear ninguna condición para que este comportamiento se activase.

En la implementación del motor, desarrolló la primera versión de la clase Place, que identifica un lugar de la historia, y la clase PlaceManager, que gestiona el conjunto de lugares y ofrece funcionalidades al resto del motor para que consulten o manipulen los lugares de la historia. También desarrolló la primera versión de la clase Character, para los personajes de la historia, y la clase CharacterController que, de igual manera al PlaceManager, gestiona los personajes y ofrece funcionalidades para consultas y gestión. Aunque las clases Place y Character no sufrieron grandes cambios a lo largo del desarrollo, a los gestores sí se les añadieron más métodos y atributos, pero más como labor de equipo que como individual, para cubrir las posibles necesidades del resto del motor.

Junto al resto de sus compañeros, terminaron de implementar las funcionalidades del motor necesarias para que la carga de los datos se hiciese de forma correcta y el prototipo de juego pudiese solicitar los datos al motor. Depuraron los errores que surgieron al conectar el prototipo con el motor y pulieron las funcionalidades del prototipo para activar y progresar las historias, planteando la carga de estas al cambiar de lugar en el juego. También implementaron que el progreso de las historias se hiciese solo cuando el personaje ha hablado con al menos un personaje de la escena y cambia de lugar, lo que permite meter varios personajes en una misma escena y que solo se pase a la siguiente si el jugador abandona el lugar.

En la implementación final del motor, la depuración y pulido de la implementación del sistema de clasificación de historias la hizo el equipo al completo. Carlos implementó las funcionalidades necesarias para que el prototipo solicitase la misión principal al motor al terminar la clasificación y las funcionalidades que decidían qué misiones secundarias se iban a devolver cuando el prototipo pidiese nuevas historias en un lugar.

Ante la casuística del recorrido de lugares en un lugar, es decir, averiguar qué lugares se recorren en una historia buscando los caminos mínimos entre escenas, se decidió que se emplearía un grafo para almacenar los lugares y se usarían algoritmos de búsqueda para calcular los caminos. Carlos buscó librerías de C# de grafos que fuesen compatibles con Unity y decidió usar QuikGraph, que incluye multitud de tipos de grafos y la mayoría de algoritmos de búsqueda, recorrido y caminos cortos estudiados.

Implementó la construcción del grafo de lugares y estudió los diferentes algoritmos que se podían usar para obtener todos los caminos mínimos del grafo. Aunque al principio se sugirió la idea de usar BFS para cada par de vértices, el algoritmo que trae la librería no almacena los caminos, sólo el estado de color (si se han visitado o no) de los vértices, por lo que no servía para el caso. Carlos vió que el algoritmo de Dijkstra es similar en complejidad a BFS cuando los pesos de las aristas es equivalente y, como el grafo no tiene pesos pues la distancia entre lugares depende del prototipo y no del motor, se podía asignar un peso de valor 1 a todas las aristas, por lo que parecía un acercamiento plausible.

De todos modos, al pensarlo un poco más, llegó a la conclusión de que se iban a repetir muchos caminos ya calculados en el grafo, lo que supondría un exceso de cómputo innecesario y costoso. Buscando una solución que aplicase programación dinámica, vió que

la librería implementa el algoritmo de Floyd-Warshall que precisamente calcula todos los caminos mínimos entre cada par de vértices de un grafo, justo el problema que se quería resolver. La implementación fue trivial y se pudo terminar de desarrollar el algoritmo de clasificación del motor.

Finalmente, la definición de las encuestas de experimentación, la búsqueda de participantes y la redacción de la memoria ha sido trabajo en equipo. A lo largo de todo el proyecto, hemos aplicado mucho la metodología de "Pair Programming" por la que se trabaja en parejas, uno escribiendo código y el otro supervisando y aportando ideas o sugerencias. Las parejas han ido rotando según la disponibilidad y la asignación de tareas, y a menudo hemos realizado reuniones en las que hemos solucionado problemas o implementado ciertas funcionalidades complejas entre todos.

11.4 Alejandro Ruiz Martín

En la primera fase del proyecto, el equipo recopila información sobre el procesado de narrativas y la narrativa de los videojuegos para sentar las bases del proyecto. También planifican algunas de las herramientas que se iban a utilizar y los lenguajes de programación apropiados para ello. En el caso concreto de este miembro, no había trabajado previamente con Unity al proceder de Ingeniería de Software, por lo que también dedicó parte de la investigación a aprender los sistemas y técnicas más frecuentes a la hora de trabajar con este motor.

Desde el comienzo el grupo también comienza a organizar los diferentes sistemas, clases y estructuras que se diseñan e implementan en el proyecto. Cada una de estas actividades es descrita por este integrante en un borrador de la memoria para que la carga de trabajo en cuanto a la documentación del proyecto no fuese muy pesada. También crea diferentes diagramas para reflejar los diseños y así lograr que el equipo tuviera más facilidad a la hora de entender cómo funcionaban los sistemas. Entre estos diagramas se encuentra una de las versiones finales del diagrama UML del motor.

Durante el desarrollo del proyecto, se fue avanzando por diferentes fases, empezando por el desarrollo del prototipo de juego, la primera versión del motor, la segunda versión de este y todas las pruebas realizadas para comprobar su correcto funcionamiento.

Comenzando por el prototipo de juego, el grupo reparte los diferentes sistemas que este contenía. El sistema asignado para Alejandro fue el de crear la clase de Interacción de la que extenderían el resto de objetos que fuesen interactivables para el jugador. En la primera versión, este sistema estaba realizado con un sistema de entrada diferente al del resto de sistemas, por lo que más tarde se adaptó para que todos los sistemas compartieran el mismo sistema. De esta clase extenderían los Objeto y los NPC.

Más adelante, tanto Alejandro como el miembro Carlos María González, fueron los encargados de desarrollar el sistema de gestión de misiones. Este sistema debía iniciar un capítulo y marcarlo como consumido para poder avanzar al siguiente. Para ello crearon dos tipos de NPC, unos denominados QuestStarterNPC, encargados de iniciar dichos capítulos y el otro tipo de NPC, el QuestFinisherNPC, encargados de finalizarlas. Estas dos clases extienden de NPC y, por ello, de la clase de Interacción, por lo que estos NPC también eran interactivables.

Al finalizar la implementación de estos sistemas y hacer pruebas básicas de funcionamiento, se comienza con el desarrollo de la primera versión del motor, que solo realiza la lectura de los ficheros de tipo json. De todos estos ficheros, desarrolla el contenido de aquellos que almacenaban la información de Objetos, Personajes, Lugares, Historias, Capítulos y Escenas, actualizándose este último al terminar de crear todos los diálogos. Algunos de estos ficheros son modificados en su estructura a la hora de añadir las nuevas funcionalidades de la segunda versión del motor.

En cuanto al motor, implementa las clases y los controladores de Item y Story, encargadas de almacenar la información de los objetos y de las historias respectivamente. El controlador de historias o StoryManager, además de encargarse de gestionar las historias, también se encargaba de manejar la información de los capítulos y las escenas que componían cada una de las diferentes historias. También participó en la creación de algunas de las funciones encargadas de enviar al juego la información que solicitase, como enviar un capítulo conociendo su identificador o enviar el capítulo siguiente.

Al finalizar la implementación del prototipo y del motor, todo el equipo procede a conectar ambos sistemas y probar las diferentes funcionalidades implementadas. Al ser estos implementados de forma independiente, surgieron algunos errores debidos a las diferencias de estos sistemas, por lo que ambos fueron ajustados para lograr la compatibilidad entre estos y lograr así el correcto funcionamiento del juego y la comunicación de este con el motor.

A la hora de realizar las pruebas con jugadores, se le asigna la creación de las primeras versiones de las encuestas, centras primero en preguntas de estadística para clasificar a los jugadores, continuando con preguntas generales sobre el juego y la cantidad de historias que lo componen y, por último, preguntas concretas sobre cada una de las historias junto con una breve descripción de esta para que fuese identificada por los encuestados. En la primera versión de las pruebas, estas últimas preguntas se centran en comprobar la relevancia de las variables de la hipótesis de partida, mientras que en la segunda experimentación estas preguntas se centran en analizar si los cálculos realizados por el motor son correctos y precisos.

En cuanto a los cálculos que realiza la segunda versión del motor, todo el equipo planteó diferentes formas de realizar los cálculos y, antes de implementar dicho sistema, realizaron cálculos de prueba para comprobar si la clasificación de historias se realizaba correctamente. Gracias a estos resultados, se pudo enfocar correctamente la fórmula encargada de clasificar las historias.

Bibliografía

- Alan Turing Scrapbook—Turing Test*. (1950). <https://www.turing.org.uk/scrapbook/test.html>
- Artificial Intelligence and Interactive Entertainment Conference*. (2007, junio 11).
<https://web.archive.org/web/20070611215121/http://www.aiide.org/speakers.html>
- ATLUS. (2016). *Persona 5 Royal*. <https://atlus.com/p5r/>
- Bethesda. (2010). *Fallout: New Vegas*. Fallout 76.
<https://fallout.bethesda.net/es/games/fallout-new-vegas>
- Bioware. (2007). *Mass Effect*.
- Blake Snyder. (, uuuu). *Save the Cat!* Dreamscape Media; WorldCat.org.
<https://bibliotecacomplutense.odilok.es/opac?id=00145082>
- Bratman, M. (1987). *Intention, Plans, and Practical Reason*. <https://doi.org/10.2307/2185304>
- Breslin, S. (2009, julio 16). The History and Theory of Sandbox Gameplay. *Game Developer (Gamasutra)*.
<https://www.gamedeveloper.com/design/the-history-and-theory-of-sandbox-gameplay>
- Brooks, K. (1995). Agent Stories. *AAAI Spring Symposium Series—Interactive Story Systems, Plot and Character*.
- Brooks, K. M. (1997). Programming narrative. *Proceedings. 1997 IEEE Symposium on Visual Languages (Cat. No.97TB100180)*. <https://doi.org/10.1109/VL.1997.626608>
- Cabrero Martín, N., García García, M., Martín García, P., Reboredo Vázquez, A., Cabrero Martín, N., García García, M., Martín García, P., & Reboredo Vázquez, A. (2021). *Sistema de gestión de narrativa en un entorno virtual controlado* [Info:eu-repo/semantics/bachelorThesis]. <https://eprints.ucm.es/id/eprint/64725/>
- Cardona-Rivera, R., & Martens, C. (2017). *Procedural Narrative Generation*.
<https://www.gdcvault.com/play/1024143/Procedural-Narrative>
- Cephalofair. (2017). *Gloomhaven*. Cephalofair. <https://cephalofair.com/pages/gloomhaven>
- Crowther, W., & Woods, D. (1976). *Colossal Cave Adventure*.
- Dehn, N. (1981). Story Generation After TALE-SPIN. *IJCAI*, 81, 16-18.

- Genette, G., & Levonas, A. (1976). Boundaries of narrative. *New Literary History*, 8(1), 1-13.
- Gervas, P. (2009). Computational Approaches to Storytelling and Creativity. *AI Magazine*, 30(3), 49. <https://doi.org/10.1609/aimag.v30i3.2250>
- Gervás, P. (2018). *LA CREATIVIDAD COMPUTACIONAL COMO FRONTERA DE LA INTELIGENCIA ARTIFICIAL Y SU POTENCIAL DE IMPACTO SOBRE LA CREACIÓN LITERARIA*. 14.
- Gold, A. (2005). *Academic AI and Video Games: A Case Study of Incorporating Innovative Academic Research into a Video Game Prototype*.
<http://nn.cs.utexas.edu/?gold:cig05>
- Gray, T. (2001, febrero 5). *CG Online—Black & White*. Computer Games.
<https://web.archive.org/web/20030525211630/http://www.cgonline.com/previews/blackwhite-01-p1-pg2.html>
- Gygax, G., & Arneson, D. (1974). *Dungeons & Dragons*.
- Interplay. (1997). *Fallout*.
- Klein, S., Aeschlimann, J. F., Balsiger, D. F., Converse, S. L., Court, C., Foster, M., Lao, R., Oakley, J. D., & Smith, J. (1973). Automatic Novel Writing: A Status Report. En *Text Processing / Textverarbeitung: Papers in Text Analysis and Text Description / Beiträge zur Textanalyse und Textbeschreibung* (pp. 338-412). De Gruyter.
<https://doi.org/10.1515/9783110837537-015>
- Lange, A. (2018, junio 15). An Intellectual History of the Sandbox. *Slate*.
<https://slate.com/human-interest/2018/06/history-of-the-sandbox-the-origins-of-a-playground-for-kids-and-ideas.html>
- Latitude. (2019). *AI Dungeon*. AI Dungeon. Recuperado 8 de septiembre de 2021, de <https://play.aidungeon.io/main/home>
- Lebowitz, M. (1983). *Creating a Story-Telling Universe*. <https://doi.org/10.7916/D8RV0WQN>
- Lionhead Studios. (2001). *Black & White*.
- Ludeon Studios. (2016). *RimWorld—Sci-Fi Colony Sim*. <https://rimworldgame.com/>
- Maxis. (2000). *The Sims*. The Sims. <https://www.ea.com/en-gb/games/the-sims/mysims>

- Meehan, J. R. (1977). TALE-SPIN, An Interactive Program that Writes Stories. *Ijcai*, 77, 9198.
- Miyamoto, S., Kondō, K., & Aonuma, E. (1986, febrero 21). *The Legend of Zelda*. The Official Home for The Legend of Zelda - Home. <https://www.zelda.com/>
- Mojang. (2009). *Minecraft*. Minecraft.net. <https://www.minecraft.net/es-es/>
- Montfort, N. (2007). *Generating narrative variation in interactive fiction*. University of Pennsylvania.
- Packard, E. (1979). *Choose Your Own Adventure*.
- People Make Games. (2020, noviembre 19). *The System Behind Hades' Astounding Dialogue*. https://www.youtube.com/watch?v=bwdYLOKFA_U
- Pérez y Pérez, R. (1999). *MEXICA: a Computer Model of Creativity in Writing*. 245.
- Perry, David., DeMaria, R. 1948-, & DeMaria, R. 1948-. (2009). *David Perry on game design: A brainstorming toolbox*. Charles River Media; WorldCat.org.
- Prince, G. (2003). *A dictionary of narratology*. U of Nebraska Press.
- Riedl, M. O. (2016). Computational Narrative Intelligence: A Human-Centered Goal for Artificial Intelligence. *arXiv:1602.06484 [cs]*. <http://arxiv.org/abs/1602.06484>
- Riedl, M. O., & Young, R. M. (2010). Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research*, 39, 217-268. <https://doi.org/10.1613/jair.2989>
- Ryan, M.-L. (2007). Toward a definition of narrative. En D. Herman (Ed.), *The Cambridge Companion to Narrative* (1.^a ed., pp. 22-36). Cambridge University Press. <https://doi.org/10.1017/CCOL0521856965.002>
- Sanderson, B., Robinette Kowal, M., Tayler, H., & Wells, D. (s. f.). *Writing Excuses*. Recuperado 19 de septiembre de 2021, de <https://writingexcuses.com/>
- Supergiant Games. (2018). *Hades*. <https://www.supergiantgames.com/games/hades/>
- Theune, M., Faas, S., Nijholt, A., & Heylen, D. (2003). The virtual storyteller: Story creation by intelligent agents. *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference, 204215*, 116.

- Togelius, J., Kastbjerg, E., Schedl, D., & Yannakakis, G. N. (2011). *What is procedural content generation? Mario on the borderline*.
https://scholar.google.es/citations?view_op=view_citation&hl=es&user=lr4I9BwAAAAJ&cstart=300&pagesize=100&sortby=pubdate&citation_for_view=lr4I9BwAAAAJ:O3NaXMP0MMsC
- Tolkien, J. R. R., Horne, M., Domènech, L., & Masera, R. (1954). *El señor de los anillos*.
- Truffaut, F. 1932-1984., Hitchcock, A. 1899-1980., Scott, Helen., Hitchcock, A. 1899-1980., & Scott, Helen. (1984). *El cine según Hitchcock* ([1a. ed., 1a. reim.]). Alianza; WorldCat.org.
- Turner, S. R. (1993). *Minstrel: A computer model of creativity and storytelling* [PhD Thesis]. University of California, Los Angeles.
- Valve. (1998). *Half-Life*. <https://store.steampowered.com/app/70/HalfLife/>
- Valve. (2004). *Half-Life 2*. https://store.steampowered.com/app/220/HalfLife_2/
- Valve. (2007). *Portal*. <https://store.steampowered.com/app/400/Portal/>
- Valve. (2011). *Portal 2*. https://store.steampowered.com/app/620/Portal_2/
- Veale, T., Amílcar Cardoso, F., & Pérez y Pérez, R. (2019). Systematizing Creativity: A Computational View. En T. Veale & F. A. Cardoso (Eds.), *Computational Creativity: The Philosophy and Engineering of Autonomously Creative Systems* (pp. 1-19). Springer International Publishing. https://doi.org/10.1007/978-3-319-43610-4_1

Anexo 1: Abstract

Narrative is a very complex field with a lot of depth. When telling a story, many variables and parameters come into play that are not visible at first glance, such as secondary narratives or small deviations from the main narrative. In the current literature and research there are not many works that focus on the computational processing of secondary narratives, focusing on the main narratives as the only narrative that affects a story. As for the automatic generation of stories, there are not many works that focus on something other than generating a main plot when, many times, stories and narratives are made up of different types of narratives or narrative arcs. Therefore, there are not many ways nor many theories about how to classify different narratives to differentiate them from each other and that can contribute to the construction of a more complex and complete narrative.

For this reason, through the study of different narrative techniques, narrative structuring and computational narrative techniques, a computational engine has been implemented that is capable of evaluating and shifting stories between main and secondary. This has been done based on the research carried out on computational narrative and classical narrative, with which a series of variables have been defined that, initially, are considered determinants for the classification. In order to prove that these variables work and are correct, a simple game prototype has been developed with stories aimed at verifying the veracity of each of the variables that make up the starting hypothesis, surveying the players after having played to check if they really affect the detection and classification of narratives. After analyzing the data obtained, it has been possible to adjust these variables and add new ones to the process, implementing an algorithm in the engine that orders the stories by applying these parameters.

Keywords

Secondary Narrative, Computational Narrative, Artificial Intelligence, Computational Creativity, Videogame.

Anexo 2: Introduction

Computational Creativity is a field of software development that deals with simulating and replicating the creative behavior of human beings, including within its fields the automatic generation and interpretation of painting, music and literature. Computational Creativity has been applied, in particular, to literature by creating different tools that are applied in various fields, such as poetry and narrative (Gervás, 2018).

For this, it has been necessary to represent computationally all the information that these fields comprise. This entire area is known as Computational Narrative, since it is responsible for the representation and processing of literary stories using software and which in turn has many things in common with Computational Creativity. It includes the areas that are responsible for telling stories, such as film scripts, novels, plays and the generation of stories in video games. In this area, it seeks, among other things, to create stories without the need for the intervention of a human being or with some human intervention, using algorithms and methods of artificial intelligence or machine learning.

Even with decades of previous study and a multitude of systems and engines capable of generating understandable stories and that follow typical narrative guidelines, the stories generated have certain limitations that make it easy to distinguish them from stories created by human beings, since the stories created by a software program cannot be equated with those created by a person. The *artificiality* of these stories created by an automated system is evident compared to the creativity of human beings, which shows how much progress remains to happen in this area.

Narrative generation systems to date are capable of creating limited stories of a certain complexity, designed to comply with a genre such as generating scripts for television dramas, or including the user in the generation itself.

In the field of video games, the narrative can be somewhat more complex if we compare it with film scripts or novels, since it is not only dedicated to telling a story, but it allows players to interact with it and *make it their own*. In other words, it allows them to influence the narrative sufficiently so that it appears that it has developed according to their designs and gives a certain sense of freedom and participation in it. This is something that gives a lot of potential to the narrative in this area since it allows the player to participate and even be responsible for his own actions, leading him to live the narrative in a totally different

way to how it is lived in other more traditional means of telling stories. (Perry et al., 2009, p. 69).

In turn, this allows a much more personal view of some stories and narratives that arise in many video games. For this reason, multiple ways of presenting them have been developed. Since there are many genres of video games, such as First Person Shooters (FPS), Roguelite / like (Games based on the procedural and random generation of content when starting a new game), Action Role Playing Games (ARPG), Point & Click (Graphic Adventures), Role Playing Games (RPG), Platforms or Strategy among others, each of these genres has its way of presenting this narrative and its way of driving it.

For example, in the video games Half Life, Half Life 2, Portal and Portal 2 (Valve, 1998, 2004, 2007, 2011, p. 2), games of the FPS genre, the protagonists, who are controlled by the player, are mute and all the characters who address them do so as if they were addressing the player directly, as if they were part of the video game, such as GLaDOS, Wheatly or G-Man, the villains who participate in these games and with whom entire communities of players have connected in their gaming sessions. This makes the player think that they are part of this diegesis and the immersion is much more complete, since this is in first person and that accentuates much more everything that happens in these adventures.

However, these games do not stop working as if they were corridors through which the player moves, completing the story that the developers have thought. Something that video games offer is the interaction of the player with the story. The power to affect in some way or another the events of the plot and build a more personal story.

This ability is shown by RPG-style games, such as Fallout: New Vegas (Bethesda, 2010) or Persona 5 (ATLUS, 2016) where they give the player to choose which paths to take. In the case of Persona 5, the player can choose between several dialogue options with variable importance in the main story, with free management of the “day to day” parts of the game and relationships with other characters, although there is usually a “correct” solution between many wrong ones. On the other hand, the case of Fallout: New Vegas is worth mentioning, since it asks the player what to do from the first minute of the game, and many ways to solve the problem with different means and results.

Despite all the freedom these titles offer, this narrative is still posed by the developer, who writes a story with a multitude of ramifications and paths that can be taken and then

presents it to the player so that they can choose where to go. But there are people who play in totally different ways, trying to make their stories as different as possible from those of the other players.

This reflects the existence of sandbox type games, which function, as the name suggests, as a playground for children. They provide a number of tools and mechanics for the player to play with and experience their narrative as they please. For example Minecraft (Mojang, 2009) does not have a narrative beyond "You appear in a world, survive and kill the dragon that hides in the dimension of The End", however by allowing the player to do what he pleases they go to generate their own narrative and will talk about the game and they will tell a story of how they managed to kill this enemy or this other one and how they got their first diamond. These types of narratives are known as emergent narratives, since they emerge as the player plays and makes a series of decisions that are not planned from the beginning by the developer.

In short, narrative generation is a very useful tool that is already used, even in its current state, to enrich the player experience and offer greater variety. There are already video games that use these techniques to generate all of their stories, such as *AI Dungeon* (Latitude, 2019).

However, in order to generate more complex narratives, it is necessary to analyze how these narratives work and what elements they are composed of. In this way, a series of algorithms and changes can be applied that would improve the generation of said narratives and improve the player experience, since a differentiation could be made between main narratives, of higher priority, and secondary narratives, which complement the experience.

From a narrative point of view, a distinction is always made regarding the importance of narrative at different levels. Starting with the most important, it's the main narrative the one that is considered the most important in the eyes of the author and the audience of the story. Usually it is usually the one that serves as an excuse to set the plot in motion and everything that it entails. For example, the plot of the destruction of the ring in *The Lord of the Rings* (Tolkien et al., 2019), is considered the main plot of the narrative that makes up the entire novel.

On the other hand, there are secondary narratives, which are those stories that complement the main narrative and normally occur at the same time but that involve less relevant or non-main characters. Continuing with the example of Tolkien's work, the entire

story of Faramir with his father would be considered a secondary narrative, since although it does not directly affect the main narrative as a whole, it does help to complement when Frodo is affected by it.

As for video games, this becomes a bit more complex because the narratives that are considered secondary may or may not affect the main narrative and may not intervene in one player's game session, while in other mediums they do. Therefore, the same parameters cannot be followed. In video games what would be considered a secondary narrative is what are known as secondary missions, since they are stories that the player decides to play.

For this reason, knowing what distinguishes the main narratives from the secondary ones in video games, and studying their characteristics, can help to implement artificial intelligences oriented to the generation of narratives with a higher level of cohesion with the diegesis of the game and that seem less artificial, since categorizing the individual stories into main and secondary allows players to know what the main plot is, commonly assigned by the developers themselves and is defined as the most important of the game's narratives, and the secondary plots, designed to complement the main one, tell less important stories or simply give the player the possibility to enjoy the game mechanics.

When this categorization is not carried out, the player's perception becomes more subjective, since many of the components that make up a story are common in main and secondary stories. Knowing the relevance that the general public gives to these components allows automated systems to create a more optimal and creative generation of stories.

The main obstacle when it comes to categorizing narratives is to represent them computationally. Since the narrative is a very abstract concept and difficult to quantify, it is necessary to divide the narrative into something more manageable for a computer and, in extension, for a program that applies Artificial Intelligence or Machine Learning techniques. For this reason, the narrative of the aforementioned games and some more was analyzed, mixing different genres to see how it was represented in them.

Focusing on the narratives raised in RPG-type games, it was possible to observe the limitations they have when proposing the narratives as missions that take the player from point A to point B and the idea of representing the stories computationally in another way was raised that would allow the program to handle them in a more modular and extensible way. For this, the idea of representing the stories and narratives in a slightly more classic way was proposed, observing the way of writing of certain novel authors who form their

stories through chapters that are in turn composed of different scenes following the structure: Story - Chapters - Scenes (Blake Snyder).

This classification allows the stories to be encapsulated in very small fragments that are more manageable by the engine and also understandable for a person, since the stories tend to work in a linear way.

However, the narratives of a video game, although they can be applied to any medium, have many more things that surround them. The term that is commonly used is *world building*, this is all the places that make up the world in which the story will take place and the characters that inhabit it, as well as cultures, past history, religions, creatures, etc. All these data influence when creating a narrative in a world, but it will not be covered in this work.

A2.1 Motivation

Given the problem of organizing a narrative for a video game, it would be very interesting to have some way or tool capable of organizing them computationally, even generating them dynamically as the player makes decisions. In addition, it is something that in the video game industry is increasingly being tested in different games and genres. For example, the video game Hades (People Make Games, 2020; Supergiant Games, 2018) already incorporates a system that activates a series of dialogues between characters based on the player's past actions, giving rise to truly entertaining and unique situations, in addition to a system that chooses these dialogues randomly while the game progresses. If these systems could be supported by another one that would help the automation of organizing the narratives or even assist in the creation of them, it could boost the development of this kind of games much more.

In addition, there are currently many tools and facilities to be able to create these tools but it has not yet been investigated enough to get to use it in games that are not experimental.

Therefore, after this approach, it makes sense and could be very useful to create a system capable of organizing the different narratives of a video game to offer a more dynamic experience to the players, starting from already written narratives that can be placed in different places on the map in order to present new stories to the player.

A2.2 Starting hypothesis

Starting from the scheme of representation of narratives raised previously, it can be observed that during the process of creating stories there are certain relevant elements involved that can lead to an accurate classification of them. Therefore, the hypothesis that arises is that a main story is one whose average length of chapters, number of places visited and number of characters, is greater than the same values for the rest of the stories that exist within a game.

Once this starting hypothesis has been raised, the influence of these values will be studied to distinguish and classify the stories and thus be able to validate or reject said hypothesis, which may undergo changes if the results are negative. To do this, a series of objectives are proposed focused on achieving the validity or rejection of the hypothesis.

A2.3 Objectives

The main objective of the project is to create a system capable of identifying secondary stories within the narrative of the game. To do this, the quantifiable parameters (number of chapters, places visited or number of characters) that make up the different stories are studied to determine which are the most decisive when identifying a story as main or secondary.

Once the parameters have been identified, it is planned to develop an engine. This engine will read the stories previously written following defined structures and will assign each one the category of primary or secondary, analyzing the parameters that have been chosen.

Following this line, the following objectives have been defined:

- Theoretical study of the operation of narrative processing.
- Analysis of the different systems created for the classification of stories.
- Design of the engine operation following the information obtained from the previous investigation.
- Search for computational variables that serve to classify stories.
- Definition of the starting hypothesis used for the development of the engine that will allow classifying the stories by measuring the variables defined in said hypothesis.

- Develop a prototype game on which to test the engine.
- Create a battery of secondary stories so that each one meets one of the variables of the starting hypothesis and a main story that meets that hypothesis.
- Implement a first version of the engine that loads the stories, previously identified, and has the necessary functionalities to be used by any game.
- Check the veracity of the hypotheses by means of a survey with a closed group of players.
- Analyze the data to discard, add, or redefine the testable values of the hypothesis.
- If this is the case, redefine the hypothesis so that it applies the new variables and thus allows the engine to categorize the stories.
- Check the correct performance of the engine by means of a survey with a different group of players.
- Analyze the results obtained.
- Draw the final conclusions based on these results.

A2.4 Methodology

At the beginning of the project, the group holds weekly meetings to study the state of the art on Computational Narrative, Computational Creativity and Narrative in general, with the aim of defining the parameters to be studied and the technologies that will be used to carry out the project. The information collected is analyzed in Chapter 2: Study of previous work.

Once the starting hypothesis and the objectives to be achieved have been defined, the development of the game prototype in Unity begins, which will serve to analyze the veracity of the hypothesis. At the same time, a first version of the engine is implemented in C# that is limited to storing the data of the stories and offering a series of functionalities to the game prototype so that it can load the stories. It is decided to increase the frequency of weekly meetings to one every two days, as it is more effective for development for group members to discuss more complex implementations and work according to the “Pair programming” model.

After the prototype is implemented, a survey is written on Google Forms and the game is published on itch.io so that a limited group of players can test it and it can be verified if the hypothesis is correct. The data obtained will be analyzed to accept or refute the hypothesis and reconsider it if necessary.

With the revised hypothesis, the implementation of the identification process that the engine will carry out is defined based on the parameters of each story. In order to carry out a second survey of a different group of players, the engine is developed and the game prototype is adapted to the changes made. Again, the full version is published on itch.io and a survey is written for the players that will be analyzed to draw the conclusions of the project.

A2.4.1 Project phases

Figure A1 is included below, which reflects the different phases of the project, together with a list of the different tasks carried out in each of these phases.

Project phase	Work description
October 2020 Project approach	Resource search: documentation for research, resources and tools. Planning: organization of the work of each phase.
November 2020 Design of the game's base systems	Systems design: gameplay and class structure. Narrative design: main story and game map. Memory: Registration of designs.
December 2020 Game development	Game implementation: class system, story management and tileset. Memory: Registration of systems and aesthetic design.
March 2021 Approach of the motor prototype	Hypothesis statement: initial study variables and design of stories that meet it. Engine design: class structure and data files with the development of the different stories. Memory: Register of designs and hypotheses.
April 2021 Implementation of the engine prototype	Engine implementation: managers and classes, file loading system and requests with Unity. Creation of files with game information. Memory: Record of engine operation.
May 2021 Experimentation and hypothesis study	Project launch and results form. Analysis of the results and readjustment of the hypothesis. Memory: Record of results and final conclusions.
June 2021 Processing engine design	Design of the classification system for histories and systems that analyze the variables of the hypothesis. Memory: Register of the systems approach.
July 2021 Engine development	Systems implementation: analysis of the variables of the hypotheses and classification of stories. Memory: Record of the history classification system.
August 2021 Experimentation and Testing	Final release with hypothesis validity form. Analysis of the data to draw a conclusion. Memory: Record of results and final conclusion.

Figure A1: Planning by project phases

A2.5 Document structure

In **Chapter 2: Study of the previous work**, we talk about the study carried out previously to be able to tackle this project. All the bibliography found for the acquisition of the necessary tools to carry out the research and the tools used for it is mentioned.

Later we proceed to explain in the sections "Game prototype design and narrative context", "Computational design", "Implementation", "Prototype tests", "Application of changes after tests" and "Final tests" all the work done. The first phase explained in **Chapter 3: Game prototype design and narrative context** and **Chapter 4: Computational design of the story processing engine** will be the development of the implementation of the system in Unity that will use the engine and prototype of this. In the development with Unity, we will talk about how the classes have been organized and the options available to the player along with an explanation of the different systems that it offers both mechanically and aesthetically.

At the end of this section, there is **Chapter 5: Implementation**, where there is a development of the internal structure of the engine together with the systems and classes in charge of storing the information from the files described in the previous chapters. It also talks about how it works depending on the different execution phases the game is in.

Once the engine operation has been defined, the tests carried out to analyze the state of the hypothesis are first developed, analyzing the variables established in it and the advances that are obtained in **Chapter 6: Prototype tests**. An explanation of how the influence of each variable of the readjusted hypothesis in **Chapter 7: Application of changes after the tests** is developed at a computational level and, finally, the new experiment is developed to verify the operation of the motor in **Chapter 8: Final tests**.

To these sections, both the performance tests of the prototype and the final version of the engine, a section of the results obtained with our discoveries and the data obtained from surveys carried out with the players who test the game will be attached.

After these tests, the results of all the tests carried out are analyzed to analyze the different data that these offer and are analyzed in conjunction with the development of the entire project in **Chapter 9: Discussion** and a conclusion based on this section and future

work applicable to this project, which are developed in **Chapter 10: Conclusions and future work**.

Finally, the individual contributions of each team member during the progress of the project are developed in **Chapter 11: Individual contribution**.

As an additional contribution, the English versions of the abstract, introduction and conclusions can be found in **Annex 1: Abstract**, **Annex 2: Introduction** and **Annex 3: Conclusions and future work**.

Anexo 3: Conclusions and future work

The research carried out on narrative in video games has allowed the development of a story identification and processing engine, which is capable of categorizing narratives into main or secondary. Several narrative components have been studied to determine their relevance in the categorization, choosing a limited number to be able to limit the experiment to the variables that we consider most remarkable: number of scenes, number of places visited and narrative importance of the characters involved.

The engine is easily scalable to take into account new narrative factors, as long as these are quantifiable and can be calculated with the existing data in the current structure. Even so, the proposed model is simple and its expansion can bring greater potential to the engine.

In future works, the data collected can be used to improve procedural generation systems that, to date, have not been oriented to introduce less relevant plots to accompany or assist the main one.

A3.1 Project conclusions

When it comes to investigating computational narrative, there is a vast documentation on the generation of stories with a single main plot, that is, narrative situations in which there is only one main story, so there would be no room for defining secondary stories. This fact contrasts with the reality in the world of video games, where the existence of secondary stories is common.

Both factors together make this research especially relevant, since on the one hand there is little information about the procedural generation of secondary narratives, and on the other hand there is little experimentation trying to generate subplots in video games.

Moving on to another of the conclusions drawn, after the first phase of the experiment that corresponds to **Chapter 6: Prototype tests**, it was observed that the variables chosen for the initial hypothesis, number of characters, number of chapters and places visited, are not the most significant for the proper functioning of an automated categorization system.

On the other hand, from the analysis of the results obtained in the second phase, it has been possible to deduce that the number of places visited is a relevant variable when it comes to categorization. In contrast, the other two factors analyzed in the first phase have been discarded, being able to be interchanged by the number of scenes and the narrative importance of the characters, which have turned out to be crucial when classifying a story as main or secondary.

On the other hand, and after the analysis of the surveys, the relevance of the characters has been highlighted by the study subjects, who also highlighted the narrative importance of the objects. Likewise, other factors have also been located, such as the ease of starting each story or the possibility of continuation at the end of the plot. All these factors extrapolated from the surveys may be new lines of future research and will be developed later in **Chapter 10.2: Future work**.

Regarding the development of the engine, the built system has been shown to be capable of structuring the stories in any project with the same technologies in which it is integrated. Due to its versatility, the use of variables to perform story processing calculations is simple and straightforward, so the inclusion of future narrative components will depend on the complexity of the factor itself to a greater extent than on the implementation in the engine.

On the other hand, the use of Unity to speed up the implementation of the prototype with which the tests have been carried out has allowed more time to be spent on research and development of the engine. Taking advantage of the model that builds the engine, the prototype has been quick to develop to be able to carry out the first test as soon as possible, and easy to adapt to the changes that were produced for the second test.

As a final conclusion, it could be noted that more research is required, both of the variables studied in this project and of those that have emerged during the analysis of results, to improve the precision of the current categorization. Along the same lines, it follows that more narrative components than those currently applied in the system are necessary to be able to deepen the processing.

Likewise, this research has tried to mark the first steps in the study of secondary video game narrative that can be expanded in future research, enhancing the generation of narrative and giving more possibilities to its application in video games.

A3.2 Future work

When studying the components and structure of the narrative, a series of variables were identified that could help to specify the identification of the stories. Due to the time constraints of the project, the team was forced to choose those considered the most relevant in order to narrow down the study and be able to make a more exhaustive analysis.

The first variable that was identified as potentially relevant is the narrative importance associated with key objects in the stories. Experimentation has proven that players found stories that included an object of great value or power to be more relevant. Some comment that, because of these objects, the story could be the main one.

This variable could be studied in a similar way to the narrative importance of the characters, assigning a numerical value by which the engine can calculate the impact it has on the story. It would be necessary to carry out more tests with specific cases to verify that it is a determining variable.

Another factor that is considered interesting to study in the future is the introduction of the story to the player, that is, how easy it is for the player to start the story. This variable has been identified after the study of the first experiment, since several respondents comment that story 1 is the main one since the character who initiates it is located just at the entrance of the protagonist's house, while other stories should be secondary as they force the player to explore or go off the main paths to start them.

This variable represents certain computational complications in the current state of the engine. Currently, the data structure of the stories contains the characters that exist in each place, but the actual position in the game, the coordinates, are foreign to the engine. Therefore, even if the distance in places between the player's starting position and the start of each story is calculated, if several starts are located in the same place, the correct data is not obtained to perform the calculation. As can be seen in figure A2, the distance in the game between the player's starting position and the characters that start stories 1, 3 and 4 is very different. However, the engine understands that the 3 characters are the same distance.

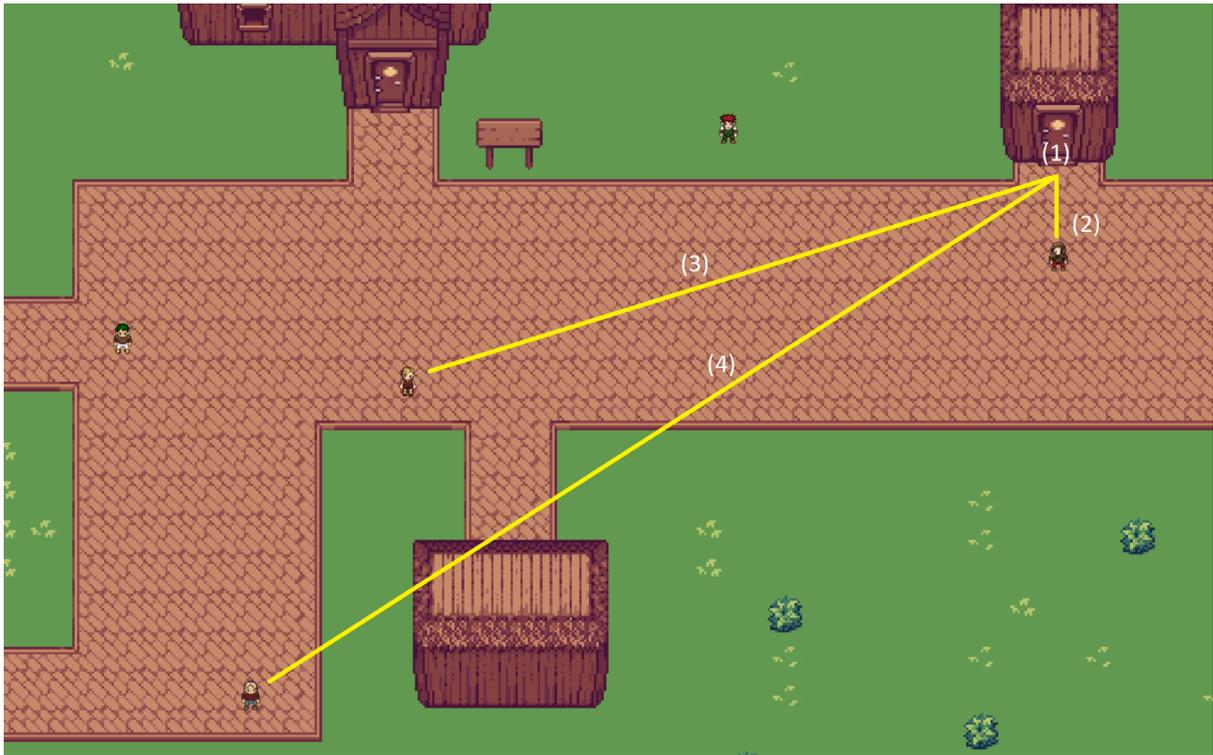


Figure A2: Image of the game prototype of the place "Fyrst", in which (1) is the position of the player when leaving his house, (2) is the distance from (1) to the character in story 1, (3) is the distance from (1) to the character in story 3 and (4) is the distance from (1) to the character in story 4.

Even so, given the results of the experiments, this factor is considered very relevant for the classification of stories and modifications to the current model are not ruled out in order to study the real impact it has on the perception of the players.

To finish with the proposals of narrative factors that have been detected during the experiment, several of the respondents commented that stories 1 and 4 could be considered main because they have open endings, that is, the plot could continue in the future even if it was not implemented in the prototype. The analysis of this variable is again complex, as there is no clear indication that the story could continue and there are many ways to leave plots open afterwards.

In addition to the analysis of variables, the possibility of establishing in the engine a "threshold" of main stories has been considered, by which several stories that had a value higher than a number were categorized as such, resulting in more than one main story in the game, either in parallel or in different situations.

Although including a value in the engine that differentiates between primary and secondary narratives may seem trivial, one must understand the nature of the calculations that the engine performs for story classification. The score for each story is obtained by studying each variable against the cumulative values of all the stories, so having a large number of stories will make the values very small and having too few stories will have the opposite result. Also, if the stories are very similar in terms of metrics, the scores could be very similar and all or none could be categorized as main.

A possible solution to assigning the delimiter value would be to calculate the threshold dynamically based on the data from the stories at each load, just as the calculation fits the data currently when calculating based on the cumulative values. Similarly, the number of main stories can be limited by studying the number of stories and their scores, ensuring that only those that are relevant are categorized as such.

If the relationship between main and secondary stories were deepened, a system could be implemented that, based on the player's situation in the main story, the secondary ones would be introduced into the game in the most natural and organic way possible, with the possible objective that the secondary narratives assist the main one. This could translate into redirecting the player to their next scene in the main one, or facilitating progress if, for example, the player is identified as having trouble overcoming the challenges (combat, puzzles, etc.) of the main one.

