

VISUALIZACIÓN INTERACTIVA DE REDES Y
COMUNIDADES EN ENTORNOS WEB
INMERSIVOS

INTERACTIVE VISUALIZATION OF NETWORKS
AND COMMUNITIES IN IMMERSIVE WEB
ENVIRONMENTS



TRABAJO FIN DE GRADO
CURSO 2022-2023

AUTOR
DAVID LAGO HERNÁNDEZ
IGNACIO URRETAVIZCAYA TATO

DIRECTOR
GUILLERMO JIMÉNEZ DÍAZ

GRADO INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

VISUALIZACIÓN INTERACTIVA DE REDES Y
COMUNIDADES EN ENTORNOS WEB
INMERSIVOS

INTERACTIVE VISUALIZATION OF NETWORKS
AND COMMUNITIES IN IMMERSIVE WEB
ENVIRONMENTS

TRABAJO DE FIN DE MÁSTER EN INGENIERÍA INFORMÁTICA

AUTOR

DAVID LAGO HERNÁNDEZ
IGNACIO URRETAVIZCAYA TATO

DIRECTOR

GUILLERMO JIMÉNEZ DÍAZ

CONVOCATORIA: JUNIO 2023

CALIFICACIÓN:

GRADO INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID
29 DE MAYO DE 2023

DEDICATORIA

A los profesores por enseñarnos las bases para realizar nuestro proyecto final, pero más en concreto a nuestro tutor Guillermo por aguantar a unas personas que tenían menos experiencia en realizar memorias que en programar antes de entrar a la carrera, y resolver aun así todas las dudas que teníamos.

AGRADECIMIENTOS

A nuestro tutor por dejarnos hacer durante 3 meses un primer prototipo sin decirnos que luego solo iba a servir para aprender.

Que ciertamente para ello sirvió y nos ayudó a mejorarlo.

A Discord por habernos proporcionado tantas horas de llamadas y tantas horas de compartición de pantalla para hacer juntos el trabajo desde nuestras casas.

A opera por proporcionar un navegador con buen rendimiento para la prueba de nuestra aplicación

RESUMEN

En los últimos años, la visualización de datos se ha vuelto crucial para comprender grandes volúmenes de información. Las redes en 3D son una opción destacada, ya que permiten representar datos complejos y multidimensionales de manera intuitiva y facilitan la identificación de patrones y tendencias.

Por otro lado, las redes son una forma efectiva de representar grupos de individuos debido a su capacidad para visualizar las relaciones y conexiones entre ellos. Proporcionan una comprensión clara de los patrones de comportamiento, la estructura grupal y las similitudes dentro del grupo. Esto facilita el análisis y la toma de decisiones informadas en diversos campos, como la sociología y la ciencia de datos.

Durante este curso se ha estado desarrollando con ThreeJS una herramienta desplegable en el navegador para la visualización en tres dimensiones de comunidades formadas por ciudadanos. Estos ciudadanos aportan contribuciones que son tratadas por algoritmos de clustering y medidas de similitud para formar las comunidades.

Para desarrollar esta herramienta, hemos implementado un motor basado en herencia que tiene el potencial de ser reutilizado en otros proyectos. Este motor proporciona una base sólida para ejecutar simulaciones con entidades y datos específicos de este caso en particular. Esto no solo agiliza el desarrollo de la herramienta actual, sino que también brinda la oportunidad de aprovechar el motor en futuros proyectos, ahorrando tiempo y recursos en el proceso.

Hemos realizado evaluaciones con usuarios, incluyendo la utilización de formularios SUS, y los resultados han sido en general favorables. Los comentarios recibidos sobre la usabilidad de la herramienta han sido buenos, lo que podría indicar que hemos logrado crear una herramienta amigable y fácil de usar. Estas evaluaciones y la retroalimentación positiva nos han brindado información valiosa para realizar mejoras adicionales y proponer otras de cara al futuro.

Palabras clave

Redes, 3D, Comunidades, ciudadanos, contribuciones, web, ThreeJS

ABSTRACT

In recent years, data visualization has become crucial for understanding large volumes of information. 3D networks have emerged as a prominent option as they allow for the representation of complex and multidimensional data in an intuitive manner, facilitating the identification of patterns and trends.

Furthermore, networks are an effective way to represent groups of individuals due to their ability to visualize relationships and connections among them. They provide a clear understanding of behavioral patterns, group structures, and similarities within the group, enabling informed analysis and decision-making in various fields such as sociology and data science.

During this course, we have been developing a deployable browser-based tool for three-dimensional visualization of communities formed by citizens using ThreeJS. These citizens contribute data, which is processed using clustering algorithms and similarity measures to form the communities.

To develop this tool, we have implemented an inheritance-based engine that has the potential for reuse in other projects. This engine provides a solid foundation for executing simulations with specific entities and data for this particular case. This not only streamlines the development of the current tool but also offers the opportunity to leverage the engine in future projects, saving time and resources in the process.

We have conducted user evaluations, including the use of SUS forms, and the results have been generally favorable. The received feedback on the tool's usability has been positive, indicating that we have successfully created a user-friendly and easy-to-use tool. These evaluations and positive feedback have provided valuable insights for further improvements and future enhancements.

Keywords

Networks, 3D, Communities, citizens, contributions, web, ThreeJS

ÍNDICE DE CONTENIDOS

Contenido

Dedicatoria	III
Agradecimientos.....	V
Resumen.....	VII
Abstract.....	VIII
Índice de contenidos	X
Índice de figuras.....	XIII
Índice de tablas	XV
Capítulo 1 - Introducción	1
1.1 Motivación	1
1.2 Objetivos.....	1
1.3 Plan de trabajo	2
1.4 Estructura del resto del documento	3
Capítulo 2 - Introduction.....	4
2.1 Motivation	4
2.2 Goals	4
2.3 Work plan	5
Capítulo 3 - Estado del arte	7
3.1 Visualización de redes.....	7
3.1.1 Dimensiones visuales.....	8
3.1.2 Visualización en 3D	8
3.2 Representación de 3D en web	9
Capítulo 4 - Contexto y metáfora visual.....	11

4.1 Contexto.....	11
4.1.1 VISIR	13
4.2 Diseño de la representación.....	16
4.3 Diseño de la interacción.....	19
4.4 Diseño de la interfaz	20
Capítulo 5 - Arquitectura e implementación.....	24
5.1 Tecnologías usadas.....	24
5.2 Motores de juego	25
5.3 Arquitectura.....	27
5.3.1 Engine	30
5.3.2 Simulator.....	31
5.3.3 Data	32
Capítulo 6 - Evaluaciones.....	34
6.1 Propósito y objetivos	34
6.2 Preguntas de investigación.....	34
6.3 Requisitos de los participantes	35
6.4 Metodología.....	35
6.5 Material de las evaluaciones con usuarios.....	36
6.5.1 Instrucciones.....	36
6.5.2 Lista de tareas	37
6.6 Resultados evaluaciones	37
6.6.2 Otras evaluaciones.....	39
6.7 Análisis de resultados.....	39
6.7.1 Tabla de análisis.....	40
6.7.2 Formulario SUS	41

6.7.3 Otros comentarios.....	42
6.7.4 Conclusiones	42
Capítulo 7 - Conclusiones y trabajo futuro	44
7.1 Revisión de objetivos	44
7.2 Trabajo futuro	45
7.3 Conclusión	45
Capítulo 8 - Conclusions and future work.....	47
8.1 Objective Review	47
8.2 Future work.....	47
8.3 Conclusion	48
Contribuciones Personales.....	49
BIBLIOGRAFÍA.....	53

ÍNDICE DE FIGURAS

Figura 3-1 force-directed layout	7
Figura 4-1 Interfaz VISIR	14
Figura 4-2 Distribución del atributo implícito	15
Figura 4-3 Ciudadanos en diferentes perspectivas	16
Figura 4-4 Habitación en los SIMS 4	17
Figura 4-5 Metáfora visual de un ciudadano	18
Figura 4-6 Filtrado de ciudadanos.....	20
Figura 4-7 Prototipo inicial de interfaz.....	21
Figura 4-8 Plano general.....	22
Figura 4-9 Información de un ciudadano.....	22
Figura 4-10 Interfaz de información de una comunidad	23
Figura 4-11 Ciudadanos sin comunidad	23
Figura 5-1 Herencia.....	26
Figura 5-2 Composición	26
Figura 5-3 Diagrama de clases	29

ÍNDICE DE TABLAS

Tabla 1 Datos de evaluación 1	38
Tabla 2 Datos de evaluación 2	38
Tabla 3 Datos de Evaluación 3	39
Tabla 4 Tabla análisis final	40
Tabla 5 Respuestas formulario SUS.....	42

Capítulo 1 - Introducción

1.1 Motivación

Para el correcto estudio y entendimiento de ciertos datos, muchas veces es apropiado una aplicación que facilite su observación. Esto se debe a que, sobre todo tras el uso de algoritmos matemáticos, el perfil de persona requerido para poder entender sus resultados, debe ser normalmente un experto en la materia o alguien que, como mínimo, este familiarizado con la información a procesar y el análisis de la misma. Esto es además un problema que solo va a ir en aumento en los próximos años debido al enorme auge de tecnologías de inteligencia artificial que son capaces de generar y tratar datos de forma tremendamente útil para los usuarios.

A pesar de estar acostumbrados a observar y analizar datos apoyándonos en modelos de visualización como pueden ser gráficos de diferentes tipos, (barras, sectores...) hoy en día es más interesante poder observar dicha información de otras formas más complejas. Estas formas más sofisticadas pueden ser redes, aportando así la información tanto de los objetos que las componen como de las relaciones entre ellos.

En términos generales, una comunidad es un grupo de personas que comparten intereses, valores, objetivos o ubicación geográfica y que se relacionan entre sí de manera regular, formando así una identidad colectiva. Es por todo esto que una representación de comunidades con forma de grafos o redes puede llegar a ser extremadamente potente, identificando a los miembros con los vértices, pudiendo modificar su forma y tamaño para representar su importancia o peso, y aprovechando las aristas para representar las relaciones entre miembros.

1.2 Objetivos

El objetivo principal de este Trabajo de Fin de Grado es el de plantear una herramienta de visualización en 3D de comunidades de ciudadanos usando redes. Par cumplir ese objetivo se han marcado estos otros objetivos secundarios:

- Investigar cómo visualizar redes en tres dimensiones y que herramientas y librerías aportarían más valor al trabajo a realizar.
- Analizar que atributos de comunidades de ciudadanos tienen sentido ser representados en 3D y que forma sería la más apropiada de modelarlos. Por otro lado, se mostrará la información no modelable en 3D, a través de una interfaz.
- Se va a generar un prototipo de la herramienta de visualización. Posteriormente se va a desarrollar un motor generalizado que pueda ser usado como prototipo de herramienta generalizado, para que permita su reutilización de cara a otros proyectos futuros.

1.3 Plan de trabajo

La ejecución del trabajo se ha llevado a cabo en base a reuniones cada 2 semanas aproximadamente en las que se comentaba y ponía al día sobre el trabajo realizado desde la anterior reunión y se fijaban metas u objetivos para cumplir de cara a la próxima.

Durante las primeras semanas del transcurso del proyecto, en septiembre, se realizaron las investigaciones pertinentes para entender qué herramientas serían más útiles en este trabajo. Más adelante, en octubre y noviembre, se comenzó a desarrollar un primer prototipo de la herramienta que se desarrollaría en el trabajo.

Este prototipo sirvió como guía para marcar un objetivo sobre el resultado final del trabajo, pero fue implementado sin un diseño arquitectónico claro y su propósito se asimilaba más al de una demo.

Finalmente, con respecto a la implementación de la versión final, esta comenzó a principios de diciembre y se ha llevado a cabo durante 6 meses hasta mayo de 2023, y se basó principalmente en refactorización de código. Además de añadir nuevas mejoras y funcionalidades.

Para el desarrollo del software se utilizó como canal de comunicación Discord, en el que los desarrolladores disponían tanto de un canal de texto como de voz, en el cual se reunían casi a diario para avanzar en el desarrollo del código. Para poder

compartir el código y utilizarlo de forma simultánea, y además de tener un sistema de versiones, se realizó un repositorio en GitHub.

1.4 Estructura del resto del documento

El Capítulo 3 - describe las distintas formas de representar los objetos y tecnologías más relevantes en este trabajo, discutiendo el estado actual de los mismos y los precedentes sobre cómo se han trabajado previamente.

Más adelante, en el Capítulo 4 - se discute el contexto sobre el que se va a trabajar, describiendo uno a uno los componentes que forman el ámbito del trabajo como son los ciudadanos, comunidades y perspectivas.

En él se explica todo lo referente a la arquitectura e implementación del proyecto a alto nivel, dividiéndose el capítulo en los diferentes componentes que lo forman. Se hace hincapié en la explicación y distinción entre el motor de juego creado y la simulación que se ha implementado por encima. A más bajo nivel y con un mayor nivel de detalle podemos encontrar la información en el Capítulo 5 - .

En él, se pueden encontrar las evaluaciones realizadas. En este capítulo se explica la metodología que se siguió, las diferentes instrucciones y lista de tareas que se les dieron a los diferentes participantes, y el análisis y conclusión final que se sacó de toda la información recopilada durante estas.

Por último, en el Capítulo 7 - se exponen las conclusiones aprendidas tras haber realizado este trabajo, extraídas tanto de la experiencia de los participantes como de la evaluación realizada a usuarios. También se discute el posible trabajo futuro de ampliación de este proyecto, de cara a aquellas funcionalidades que no ha sido posible incluir.

Capítulo 2 - Introduction

2.1 Motivation

For the proper study and understanding of certain data, it is often appropriate to have an application that facilitates its observation. This is because, especially after using mathematical algorithms, the profile of a person required to understand the results should typically be an expert in the field or someone who is at least familiar with the information being processed and its analysis. This is also a problem that will only increase in the coming years due to the tremendous rise of artificial intelligence technologies capable of generating and processing data in a highly useful manner for users.

Despite being accustomed to observing and analyzing data using visualization models such as various types of charts (bars, pie charts, etc.), nowadays it is more interesting to be able to observe this information in more complex ways. These more sophisticated forms can be networks, providing information about both the objects that compose them and the relationships between them.

In general terms, a community is a group of people who share interests, values, goals, or geographical location and regularly interact with each other, thus forming a collective identity. It is for all these reasons that a representation of communities in the form of graphs or networks can be extremely powerful, identifying members as vertices and allowing for modifications in their shape and size to represent their importance or weight, while utilizing edges to represent the relationships between members.

2.2 Goals

The main objective of this Final Degree Project is to propose a 3D visualization tool for communities of citizens using networks. To achieve this objective, the following secondary objectives have been set:

- Investigate how to visualize networks in three dimensions and determine which tools and libraries would provide the most value to the work being done.
- Analyze which attributes of communities of citizens make sense to be represented in 3D and determine the most appropriate way to model them.

Additionally, non-modelable information will be displayed through an interface.

- Generate a prototype of the visualization tool. Subsequently, develop a generalized engine that can be used as a prototype for a generalized tool, allowing for its reuse in other future projects.

2.3 Work plan

The work was carried out based on meetings held approximately every two weeks, where progress since the previous meeting was discussed and goals or objectives were set for the upcoming period.

During the early weeks of the project, in September, relevant research was conducted to understand which tools would be most useful for this work. Later, in October and November, the development of an initial prototype of the tool began, which would be further developed throughout the project.

This prototype served as a guide to establish a final goal for the project, but it was implemented without a clear architectural design, resembling more of a demo in its purpose. Regarding the implementation of the final version, it started in early December and was carried out for six months until May 2023, focusing mainly on code refactoring. In addition to adding new improvements and functionalities.

Discord was used as the communication channel for software development, providing developers with both text and voice channels. They would gather almost daily to make progress on the code. To facilitate code sharing and simultaneous usage, as well as version control, a repository was set up on GitHub.

Capítulo 3 - Estado del arte

Antes de empezar a trabajar sobre la aplicación, se realizaron una serie de investigaciones relacionadas con todo lo que se va a utilizar para desarrollarla. En este capítulo se expone una explicación sobre las investigaciones que se realizaron a cerca de la representación de grafos o redes y sobre la representación de 3D en web.

3.1 Visualización de redes

Una red o grafo es una estructura que representa un conjunto de vértices y la relación que se establece entre ellos, llamadas aristas.

La forma más común de representar una red es modelando círculos para representar los nodos, y líneas para unirlos de forma correcta para representar las aristas. El problema que tiene esta representación es cuando la red es muy grande, ya que sería muy complicado diferenciar información.

La solución a esto se puede conseguir mediante la técnica *force-directed layout* que consiste en la asignación de fuerza a cada nodo y arista para poder permitir que se recalculen iterativamente las posiciones de los nodos en función de la influencia de los demás, combinando atracción y repulsión. Después de n iteraciones el sistema empezará a estabilizarse y mostrar una posición más estable. De esta forma se revelan estructuras como comunidades.

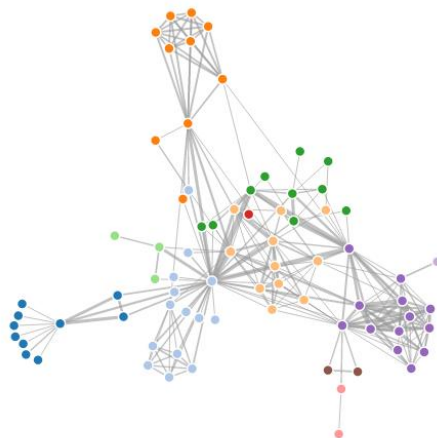


Figura 3-1 force-directed layout

En la Figura 3-1, como se cita en un artículo de la UOC [1]: *Este ejemplo, de Mike Bostock, muestra un grafo de red representando las relaciones entre los diferentes personajes de la obra de Víctor Hugo Los miserables. Cada nodo representa a un personaje y la línea que une dos personajes indica que dichos personajes están relacionados por apariciones simultáneas en la obra. El color de cada nodo se corresponde al capítulo en el cual aparecen. El grueso de cada arista muestra el volumen de apariciones simultáneas de los dos personajes.*

3.1.1 Dimensiones visuales

Para representar la información de forma correcta y más extensa en las redes se pueden utilizar diferentes dimensiones visuales. Las más comunes son:

- **Tamaño:** Los diferentes tamaños tanto en los nodos como en las aristas pueden representar un atributo relevante, y normalmente suele representar la importancia de un nodo o arista.
- **Color:** Para representar otros atributos, normalmente categorías, se suelen emplear los colores, tanto en los nodos como en las aristas. De esta forma, se puede rápidamente diferenciar.
- **Forma:** Otra dimensión muy utilizada es la forma, y al igual que el color, normalmente se suele utilizar para representar atributos categóricos.

3.1.2 Visualización en 3D

Para representar redes más avanzadas, visualizar en 3D se ha convertido en una de las técnicas más conocidas y eficaces, ya que te permite ampliar las dimensiones visuales y añadir una perspectiva. Algunos de los aspectos más importantes son:

- **Perspectiva:** Gracias a la tercera dimensión se nos permite agregar profundidad y dar una mejor perspectiva a nuestra red. Gracias a esto los nodos y las aristas pueden situarse en diferentes planos y ayudarnos a comprender mejor la información de las relaciones que tienen entre ellos.

- **Interacción:** La visualización en 3D permite una mayor interacción con la red, ya que te puedes mover en los diferentes planos y rotar para conseguir una perspectiva diferente.
- **Dimensiones visuales:** Gracias a la visualización en 3D, el campo de dimensiones visuales se dispara exponencialmente, ya que puedes cargar modelos con diferentes texturas y tamaños, y esto añade una gran cantidad de posibilidades a la hora de representar diferentes atributos.

3.2 Representación de 3D en web

Para la representación de modelos en 3D se puede encontrar la tecnología WebGL que está basada en OpenGL ES, que permite realizar renderizados gráficos en navegadores web utilizando JavaScript y HTML5. Además, WebGL aporta una gran calidad visual y un rendimiento óptimo.

Entre los archivos más populares para la representación de los modelos 3D en la web se encuentran:

- **OBJ**, un formato de archivo muy básico en el que se almacena la geometría de los modelos y también sus texturas.
- **FBX**, que es un formato desarrollado por Autodesk, y que contiene además de todo lo de OBJ, también contiene materiales y animaciones, además de más utilidades, aunque es más complejo que OBJ, es mucho más completo y útil.

Para representar estos modelos, existen varias librerías, entre ellas, la que se va a usar en el proyecto, ThreeJS, que utiliza un renderizado a través de WebGL. Además, se investigó otra antes de decantarse por esta, BabylonJS.

- **BabylonJS** se enfoca en proporcionar una solución todo en uno para la creación de juegos y aplicaciones 3D, incluyendo motores de física, herramientas de edición de escenas y soporte para dispositivos de realidad virtual y aumentada.

- **ThreeJS** se enfoca más en proporcionar una API flexible y de bajo nivel para trabajar con gráficos 3D, lo que permite una mayor personalización y control para desarrolladores.

Otra diferencia importante es la comunidad de usuarios y la documentación disponible. ThreeJS ha sido una biblioteca muy popular durante muchos años, lo que significa que hay una gran cantidad de recursos y tutoriales disponibles en línea, así como una comunidad activa de desarrolladores. BabylonJS también tiene una comunidad activa y en crecimiento, pero no es tan grande como la de ThreeJS.

Capítulo 4 - Contexto y metáfora visual

Este trabajo se engloba dentro de un proyecto europeo en el que es necesario visualizar tanto comunidades de ciudadanos cómo las relaciones entre ellos y el motivo por el que se forman dichas comunidades. Para ello, ya hay una herramienta para la visualización en dos dimensiones que permite observar las relaciones mencionadas y la distribución de un grupo de ciudadanos en función a ciertas medidas de similitud.

De este modo, en este capítulo se introducirán tanto los conceptos de comunidad, ciudadano y perspectiva, para poder entender el contexto del proyecto, como el concepto de contribución, clave para la formación de comunidades. También se hará un breve repaso a la herramienta en dos dimensiones ya que esta ha servido de gran apoyo para el desarrollo de este Trabajo de Fin de Grado.

4.1 Contexto

Este Trabajo de Fin de Grado se engloba dentro del proyecto europeo SPICE¹, que explora la idea de animar a los ciudadanos a desarrollar y compartir su interpretación personal sobre piezas culturales y artísticas. El fin de este proyecto es el de mejorar la cohesión social y promover la participación inclusiva de diferentes grupos de ciudadanos que interaccionan con un legado cultural y artístico. Además, el proyecto SPICE pretende plasmar la diferencia entra el sentimiento de pertenencia que puede tener un ciudadano con un grupo, y la clasificación que se le puede llegar a dar a ese ciudadano en base a sus reacciones ante distintas obras de arte. Estas comunidades les sirven de utilidad a los ciudadanos para reflexionar en su identidad como individuo y ayudan a interpretar las obras de un modo diferente al esperado.

En su significado más crudo, las comunidades de ciudadanos se entienden como grupos de personas que tienen tanto una identidad como un sentido de pertenencia, formado por las comunidades a las que pertenecen. Por otro lado, como se mencionó

¹ Social cohesion, Participation, and Inclusion Through Cultural Engament. Proyecto financiado por el programa de investigación e innovación de la Unión Europea H2020 (Grant Agreement N. 870811)

previamente, un ciudadano puede ser clasificado en una comunidad diferente en base a sus contribuciones.

Un ciudadano se entiende como una entidad que registra contribuciones con obras de arte y que reacciona a ellas. La emoción o sentimiento extraída de esa reacción es importante para entender como se ha sentido ese ciudadano con respecto a una obra de arte concreta. Cada uno de los ciudadanos tiene una serie de atributos, llamados explícitos, por ejemplo, el género o edad. Dichos atributos definen al ciudadano y lo dotan de profundidad para poder diferenciarlo del resto de ciudadanos.

Las contribuciones, siendo estas las reacciones que tienen los ciudadanos a una obra de arte concreta son claves ya que son usadas para determinar grados de similitud entre ciudadanos basándose en la emoción o sentimiento extraída de la interacción con la obra de arte. Estas similitudes pueden ser usadas para crear comunidades de ciudadanos que hayan generado contribuciones similares, ya sea a nivel de similar reacción ante una pieza de arte o hayan reaccionado a obras de artes similares.

En consecuencia, una vez procesados estos datos se generan comunidades de ciudadanos cuyas contribuciones han sido relacionadas, formando así una comunidad. Para la formación de dicha comunidad se habrán tomado en cuenta ciertas contribuciones de los ciudadanos, mientras que otras no habrán sido relevantes

Esto es interesante ya que de esta disparidad de criterios entre algoritmos nace el concepto de perspectivas. Una perspectiva es el resultado final del proceso que forma las comunidades a las que pertenecen los ciudadanos en base a sus contribuciones usando unas determinadas medidas de similitud y un determinado algoritmo de detección de comunidades. De acuerdo con esto, las comunidades generadas usando una perspectiva A no tienen por qué ser las mismas que las que se generan usando una perspectiva B. Por lo tanto, un mismo ciudadano puede pertenecer a distintas comunidades en A y en B.

4.1.1 VISIR

VISIR (VISualization for Interpretation and Reflection) es una herramienta de visualización de comunidades creada para el proyecto SPICE² con el fin de dar soporte al SPICE-IRL (Interpretation-Reflecion Loop). Es la herramienta encargada de detectar y representar las comunidades que emergen a partir de las interpretaciones de los ciudadanos de obras culturales o de arte.

Esta visualización permite analizar la distribución de ciudadanos, patrones locales, e incluso huecos con respecto a sus emociones al reaccionar a ciertas obras de arte. VISIR permite la exploración de diferente perspectiva, ofreciendo así la posibilidad de comparar e interpretar las configuraciones desde diferentes puntos de vista. Esta es la funcionalidad más potente de VISIR ya que aprovecha de lleno la posibilidad de que con medidas de similitud y algoritmos diferentes el mismo ciudadano pueda pertenecer a dos comunidades distintas.

La interfaz de VISIR se presenta en la Figura 4-1, en la que se puede ver como se muestran dos perspectivas diferentes, cada una con sus comunidades y ciudadanos. Una perspectiva es representada por un grafo en el que los nodos son los ciudadanos, y los enlaces representan las similitudes entre los ciudadanos. En el caso concreto de la **¡Error! No se encuentra el origen de la referencia.**, los atributos implícitos son representados con la forma del nodo y su color. La forma representa el interés en ver contenido con lenguaje de signos, y el color del nodo representa el interés que tiene el ciudadano en el arte.

² <https://spice.fdi.ucm.es/visir/>

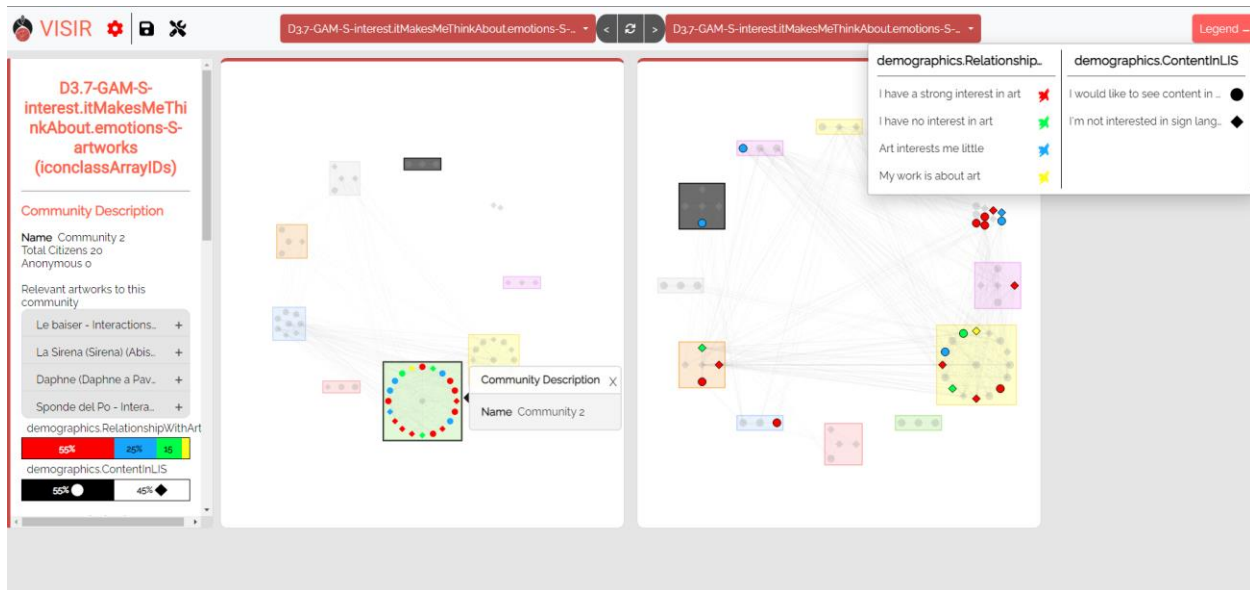


Figura 4-1 Interfaz VISIR

Aquí, una comunidad es representada por un rectángulo que contiene una serie de nodos, en este caso los ciudadanos que pertenecen a la comunidad representada por ese rectángulo. Los nodos que no se encuentran dentro de ningún rectángulo son ciudadanos que el algoritmo ha no ha sido capaz de configurar dentro de ninguna comunidad.

Cuando el usuario pulsa sobre una comunidad se abre un panel como el izquierdo en la Figura 4-1. En este panel aparece información para poder reconocer la comunidad, como el nombre de la misma, el número de ciudadanos y la cantidad de ciudadanos anónimos que pertenecen a ella. Además, aparecen las obras de arte más relevantes para esta comunidad, es decir, las obras de arte que están más presentes en las contribuciones de los ciudadanos. Se despliega también información sobre los atributos explícitos, ofreciendo porcentajes sobre la cantidad de ciudadanos que tienen interés tanto en el arte como en ver información de lenguaje de signos. Por último, en lo referente a este trabajo aparece una distribución de los atributos implícitos, es decir, de las emociones registradas en las contribuciones que caracterizan a esa comunidad, como se puede ver en la Figura 4-2.

Percentage distribution of the implicit attribute (interest.itMakesMeThinkAbout.e motions): : interest.itMakesMeThinkAbout.e motions



Figura 4-2 Distribución del atributo implícito

Por otro lado, cuando se pulsa sobre uno de los nodos, el panel informativo que se abre muestra en primera instancia los atributos implícitos del ciudadano, en este caso su interés en el arte y su interés con el lenguaje de signos. Posteriormente aparecen sus contribuciones que han determinado que pertenezca a su comunidad, y por último aparece la misma información sobre la comunidad que cuando se pulsa sobre una de ellas.

Además, VISIR ofrece también la opción de aplicar filtros a las perspectivas para poder ocultar o restar importancia a ciertos elementos en caso de que se quiera prestar más atención a otros. Esta funcionalidad está implementada en la leyenda visible en la esquina superior derecha de la Figura 4-2, en la que al pulsar sobre uno de los elementos de la leyenda oculta a los ciudadanos caracterizados por el rasgo pulsado.

Por último, para apoyar la labor de comparación y estudio de la información mostrada, en VISIR, si se tienen 2 perspectivas abiertas al mismo tiempo y se selecciona un ciudadano en una de las perspectivas, este ciudadano será también resaltado en la otra perspectiva para poder comparar qué comunidad se le ha asignado en base a las diferentes medidas de similitud, como se puede ver en la Figura 4-3.

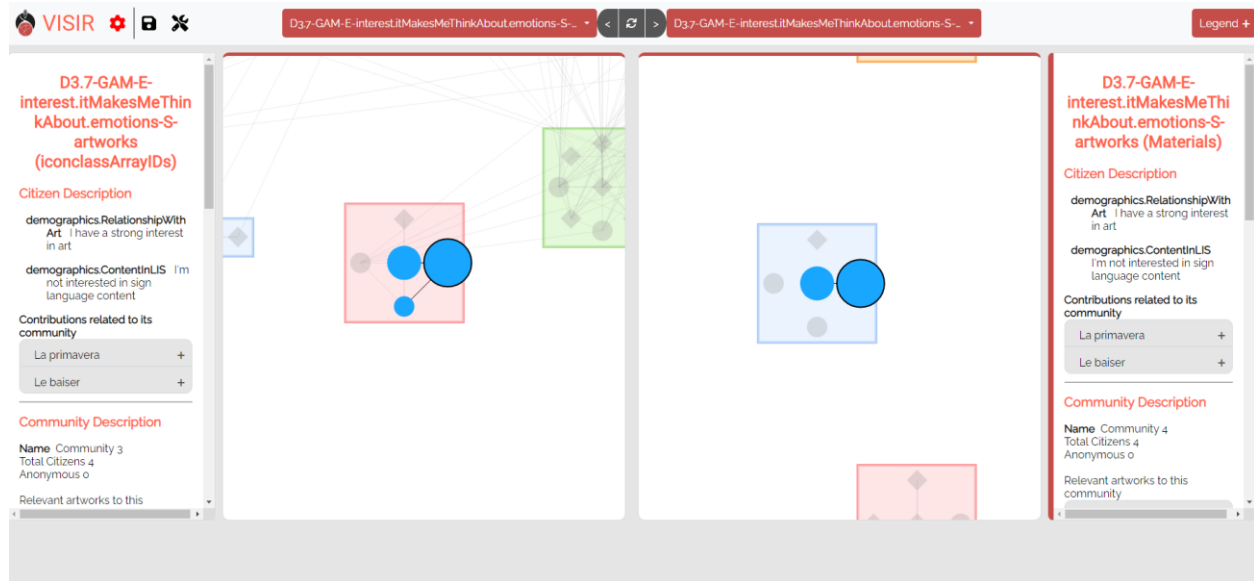


Figura 4-3 Ciudadanos en diferentes perspectivas

4.2 Diseño de la representación

Ya que el propósito de este Trabajo de Fin de Grado es el de desarrollar una herramienta de visualización 3D de comunidades formadas por ciudadanos que registran contribuciones, y VISIR trabaja bajo el mismo contexto se va a utilizar la herramienta VISIR como punto de partida para el diseño de la herramienta.

Al tratar el trabajo sobre la representación de las ya explicadas perspectivas, lo primero fue estudiar cómo se iba a representar una perspectiva, con vistas a un futuro en el que se pudieran ver diferentes perspectivas al mismo tiempo. Esto llevó a la idea de que una perspectiva tenía que ser de alguna forma modular, en el sentido de que tenía que poder verse en tres dimensiones de forma clara, con un principio y un final visibles. Por ello se decidió representar una perspectiva concreta como una habitación, en la que habitarían los ciudadanos, organizados en sus comunidades.

Para representar dicha habitación principal de nuestro trabajo, se utilizó como ejemplo la representación de una habitación como se hace en el videojuego Los Sims³. Es decir, tener una visión desde un punto considerablemente elevado para observar la habitación, y que según donde esté situada la cámara, las paredes de la habitación se hagan invisibles para que no quiten visión como indicamos en la Figura 4-4;**Error! No se encuentra el origen de la referencia..**

En cuanto a la disposición y organización de la habitación, se ha optado por situar un polígono regular imaginario cuyo número de vértices es el número de comunidades que se quieren representar en esa perspectiva en el centro de la habitación, de tal forma que cada una de las comunidades de esa perspectiva se sitúa en cada uno de los vértices del polígono regular.



Figura 4-4 Habitación en los SIMS 4

Con respecto al apartado de la representación de ciudadanos, hubo dos aproximaciones distintas que se consideraron. La primera era utilizar los mismos colores y formas que en VISIR, pero con modelos en 3 dimensiones. La segunda (la escogida finalmente) trataba de utilizar modelos en 3 dimensiones de personas para representar algunos de los atributos explícitos (por ejemplo, modelos de niños y adultos para representar la edad).

Como ya se ha comentado los ciudadanos cuentan cada uno con una serie de atributos explícitos que idealmente contarían todos ellos con su representación visual tridimensional. Ya que esto no es posible, lo que se ha decidido centrarnos en unos

³ <https://www.ea.com/es/games/the-sims/the-sims-4>

pocos parámetros de cara a la representación en tres dimensiones: el género del ciudadano y su edad.

En nuestro trabajo hemos decidido representar el género del ciudadano con su modelo en tres dimensiones, estableciendo únicamente dos géneros, hombre y mujer, para evitar las complicaciones y compromiso que puede generar el representar un concepto como ese. La edad del ciudadano se representa de la misma manera con el modelo dentro de la perspectiva, diferenciando entre jóvenes, adultos y ancianos. Por para modelar la lengua que hablan los ciudadanos ha sido, se ha utilizado como metáfora visual un billboard en el que se muestra la bandera del país de la lengua. Un ejemplo de la metáfora visual completa para el ciudadano se ve en la Figura 4-5.



Figura 4-5 Metáfora visual de un ciudadano

En cuanto a la representación de comunidades se ha optado por una aproximación muy similar a la de VISIR, agrupando a los ciudadanos pertenecientes a una misma comunidad en un círculo delimitado en un principio por una línea, y más adelante en por una tarima saliente del suelo de la habitación. Profundizando más sobre

el porqué de esta decisión, un ciudadano es fácilmente identificable y tiene su modelo tridimensional. Sin embargo, una comunidad, no lo es tanto. Ya que la comunidad es un concepto abstracto al ser un grupo de personas, se vio que se tenía que añadir un elemento tangible y visible para poder hacer alusión al inicio y fin de una comunidad: la tarima. Los ciudadanos de cada comunidad se sitúan encima de cada tarima formando círculos concéntricos. Es decir, un ciudadano en el centro de la comunidad y los demás formando círculos de radio cada vez mayor alrededor del primero.

4.3 Diseño de la interacción

La interacción con los objetos en tres dimensiones es una parte fundamental de la aplicación, debido no solo a las ya mencionadas limitaciones que se tienen para representar ciertos atributos de los ciudadanos, sino también a la necesidad de poder observar los detalles de las comunidades y la información que cada una de ellas contiene.

De este modo, se puede tanto hacer doble clic tanto en los ciudadanos como en las comunidades para examinarlos de cerca. Al seleccionar una comunidad, aparecerá un borde coloreado alrededor de su tarima y se hará zoom a ese grupo en concreto, para que se pueda observar en detalle a todos los ciudadanos que la forman. Acompañado de este zoom, se abrirá una ventana en la esquina superior derecha con la información de la comunidad, que se describirá en la siguiente sección.

De la misma forma para los ciudadanos, al hacer doble clic sobre uno de ellos se hará zoom sobre el ciudadano y se elevará y agrandará su modelo tridimensional para que el usuario pueda observar en primer plano el ciudadano seleccionado. Por último, el clic sobre el ciudadano también viene acompañado, como no podía ser de otra manera, de la apertura de la ventana de información para poder tener acceso así a sus atributos explícitos que no se han representado en tres dimensiones.

Además, al igual que en VISIR se implementa una leyenda con funcionalidad de filtro, en la parte inferior derecha se puede realizar un filtrado accionando los diferentes botones. De tal manera que los que no cumplan las características tendrán una opacidad inferior. Este resultado y la opción de filtrado se puede ver en la Figura



4-6 Error! No se encuentra el origen de la referencia..

Figura 4-6 Filtrado de ciudadanos

Sobre el aspecto del filtrado, se ha puesto como límite en el cuadro inferior derecho el de 3 atributos con 4 valores cada uno, por lo que, si se desearan añadir más atributos, no se podría filtrar a través de ellos. Lo que sí se podría hacer es añadir, por ejemplo, otro valor al atributo de edad o género.

4.4 Diseño de la interfaz

La interfaz de usuario es una pieza fundamental de la aplicación ya que a través de ella se va a tener acceso a la mayoría de los datos e información sobre los ciudadanos, sus interacciones con el arte y sus datos y los datos de las comunidades. Además, al estar este trabajo basado en VISIR, consideramos también el añadido de una leyenda que aporte información y funcionalidad como el filtrado.

En cuanto a la distribución y estilo de la interfaz de usuario de la aplicación, surgieron varias ideas para tratar de ser lo menos intrusiva posible y así aprovechar al máximo la pantalla.

En un principio aparecieron tres propuestas diferentes. Sin embargo, estas contaban con una serie de opciones y funcionalidades que finalmente no se van a incluir en la versión final del trabajo por lo que muchos de los problemas que encontramos fueron desapareciendo. Algunas de estas opciones eran la disposición de la representación, el archivo origen de los datos, u otros parámetros como el grado de similitud entre ciudadanos para mostrar aristas entre ellos, una función que se desechó para este trabajo. Una de estas propuestas se puede ver en la Figura 4-7.

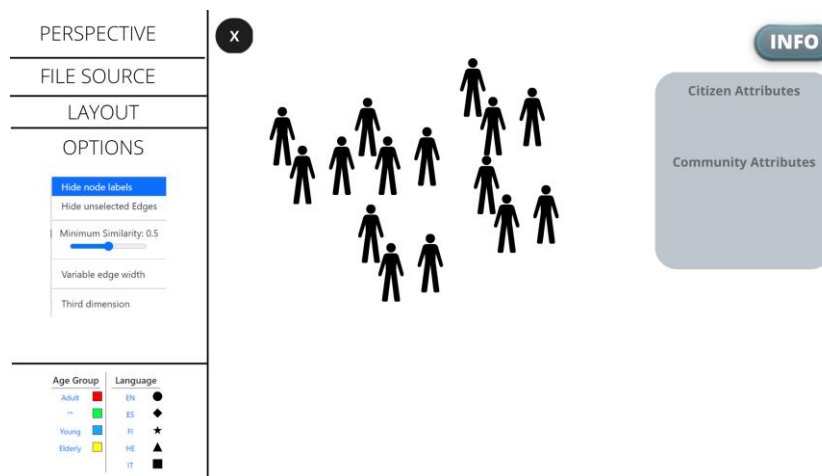


Figura 4-7 Prototipo inicial de interfaz

Finalmente se ha continuado con la línea del diseño menos intrusivo posible, aportando la interfaz en un principio tan solo dos cuadrados pequeños en ambas esquinas derechas de la pantalla como se puede ver en la Figura 4-8. El icono de información superior derecho se amplía cuando se pulsa sobre un ciudadano o una comunidad y muestra su información. Por otro lado, el inferior derecho sirve para poder filtrar en función a los atributos de los ciudadanos. Para abrir este cuadro, basta con hacer un clic sobre el botón.



Figura 4-8 Plano general

Para el ciudadano se ha mostrado toda su información explícita, en nuestro ejemplo la edad, género e idioma, además de las ya representadas visualmente. Además, se ha proporcionado una imagen de la obra de arte, que se atribuye a la contribución más importante de ese ciudadano a la comunidad a la que pertenece. Se puede observar un ejemplo en la Figura 4-9.

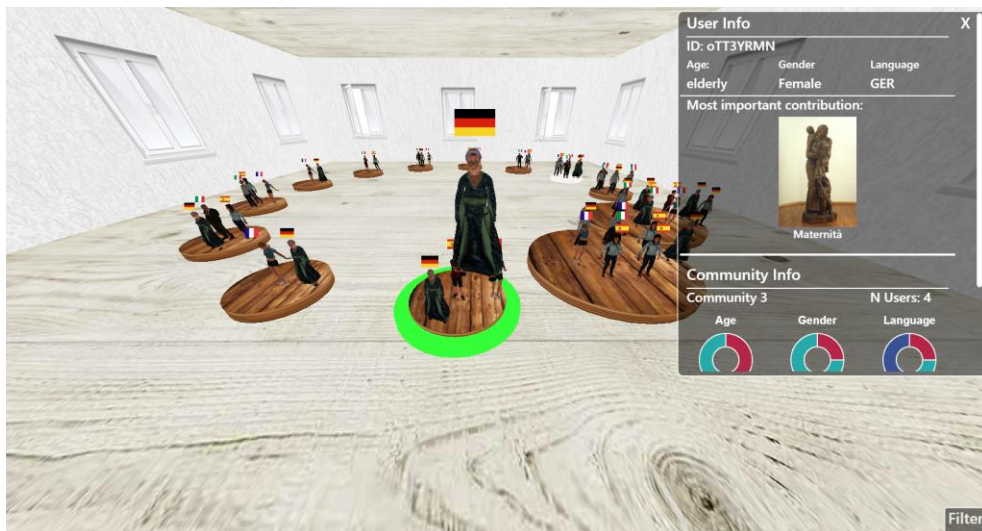


Figura 4-9 Información de un ciudadano

Para las comunidades se ha dado tanto la información explícita de sus ciudadanos, creando unas graficas de ellas en las que se indica el porcentaje de cada uno de los atributos correspondientes dentro de la comunidad. También se muestra la información implícita de la propia comunidad junto a su nombre y el número de ciudadanos que la componen. Se puede observar un ejemplo en la Figura 4-10.

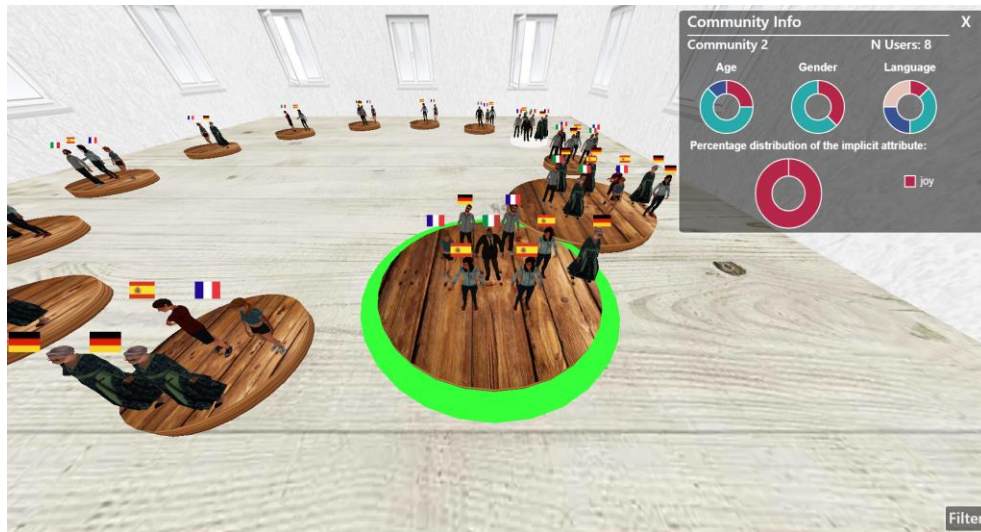


Figura 4-10 Interfaz de información de una comunidad

Además, se ha representado a un grupo de ciudadanos sin comunidad, como una comunidad especial, en el que la tarima en la que se sostienen es blanca transparente, para así diferenciarlos. De la misma forma al hacer clic sobre ella, se mostrarán los atributos porcentuales, pero no los implícitos, ya que estos usuarios no comparten estas características. Se puede ver un ejemplo en la Figura 4-11.



Figura 4-11 Ciudadanos sin comunidad

Capítulo 5 - Arquitectura e implementación

En este capítulo se va a hablar con detalle de cómo se va a realizar toda la aplicación, tanto el diseño arquitectura como la implementación final de la aplicación, dando a conocer las diferentes opciones que se han manejado a lo largo de la misma, y sus correspondientes decisiones finales.

La aplicación ha sido diseñada para mostrarse y ejecutarse en navegadores Web es por ello por lo que se ha usado JavaScript como lenguaje de programación y NodeJS, ya que tenemos experiencia y hemos trabajado con anterioridad.

Para el correcto desarrollo y mejora en la aplicación se desarrollaron dos versiones, una primera, sin una estructura ni una arquitectura clara, pero con una funcionalidad correcta, en la que se puede ver el código en GitHub⁴.

Y una segunda iteración con una estructura mucho más compleja, pero a su vez mucho mejor implementada. En la que se puede ver el código en otro repositorio de GitHub⁵. Esta segunda iteración se ha desplegado en la página de nuestro repositorio⁶ a través de GitHub Pages

5.1 Tecnologías usadas

Para realizar la empaquetación y despliegue de la mejor forma la aplicación nos hemos ayudado de Webpack [2] y gracias a ello, se ha desplegado de forma más sencilla en GitHub Pages la aplicación para su correcta visualización en los navegadores sin la necesidad de realizar ningún tipo de instalación.

Por las razones mencionadas anteriormente en el capítulo 3.2, y realizando un estudio del uso que va a tener la aplicación, se decidió utilizar ThreeJS, ya que resultará más sencilla y cómoda y nos proporcionará todo lo necesario para representar y realizar correctamente todo lo esperado.

⁴ <https://github.com/davлаго/TFG-Web-3D>

⁵ <https://github.com/davлаго/TFG-VISIR>

⁶ <https://davлаго.github.io/TFG-VISIR/>

Para la integración de gráficos, y con ello mostrar mejor una serie de estadísticas, se miraron dos librerías: D3.js [3] y Chart.js [4]. D3.js es una librería muy potente y demasiado compleja para lo que se necesitaba. Sin embargo Chart.js es mucho más sencilla y fácil de implementar. Con lo que para los pocos y sencillos datos que se querían usar y mostrar, se decidió usar Chart.js.

También se realizó, en la primera iteración, el estudio de Docker para realizar el despliegue, y se dejó implementado. Sin embargo, para la segunda iteración se buscó, como se ha mencionado antes, la comodidad de usuario, con lo que gracias a Webpack y GitHub Pages se consiguió un despliegue inmediato y sencillo para el usuario.

5.2 Motores de juego

Se ha decidido implementar un motor de juego, y para saber en qué arquitectura basar nuestra aplicación se realizó una investigación para encontrar la mejor solución para la implementación. Como guía se usó de referencia el libro [5] en la que se estudiaron dos opciones: arquitectura basada en herencia y arquitectura basada en componentes.

- La **herencia** es un mecanismo mediante el cual una clase (llamada clase padre) que es utilizada como base para crear una nueva clase (llamada clase hijo). La clase hijo hereda todas las propiedades y métodos de la clase padre y puede agregar o modificar su propio comportamiento. En un motor de juego, la herencia se puede utilizar para definir clases padre para objetos comunes, como personajes, enemigos, objetos, etc., y luego crear clases hijo que agreguen comportamiento específico. Un ejemplo podría ser la construcción de un coche y de varios tipos de coches, en este caso, por herencia, se vería que la clase padre tiene las propiedades y las clases hijas van heredando y usando los métodos del padre como se muestra en la Figura 5-1.

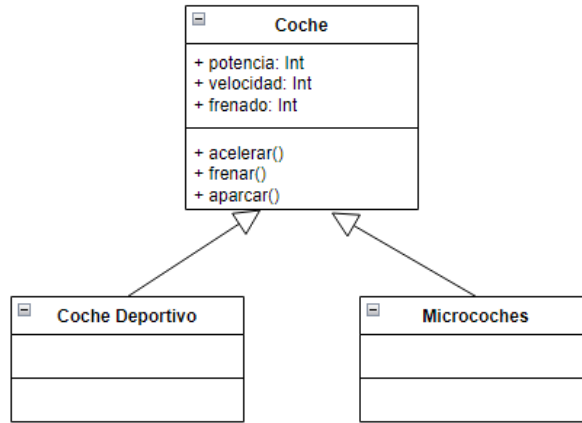


Figura 5-1 Herencia

- La **composición** es un mecanismo en el que el objeto principal se compone de uno o más objetos de otras clases. En lugar de heredar propiedades y métodos, el objeto contiene a otros objetos y manda realizar el comportamiento a ellos. En un motor de juego, la composición se puede utilizar para definir componentes comunes, como renderizado, física, sonido, etc., y luego agregar esos componentes a objetos específicos según sea necesario. Un ejemplo utilizando el dado en la herencia, sería elaborar una clase coche, y una serie de componentes, motor, ruedas, etc., y que la subclase sea la que añada los componentes que necesita a la clase principal n ejemplo como se muestra en la Figura 5-2.

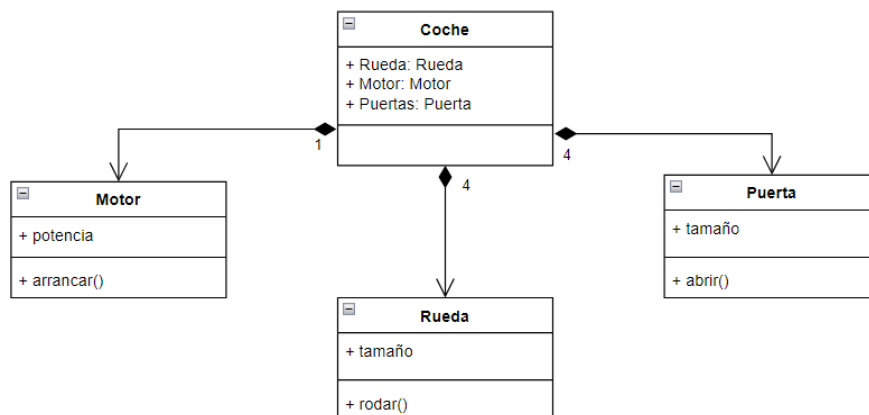


Figura 5-2 Composición

En general, la herencia se utiliza para definir jerarquías de clases y comportamiento común, mientras que la composición se utiliza para definir unos componentes reutilizables y comportamiento modular. En un motor de juego, los dos tipos pueden ser útiles y se pueden utilizar juntos incluso para lograr una buena implementación, además de hacerla fácilmente escalable. Sin embargo, es importante tener en cuenta que la herencia puede conducir a una jerarquía de clases con mucha profundidad, mientras que la composición puede generar una estructura muy compleja si se elaboran muchos componentes y clases distintos. Por lo tanto, lo mejor es saber encontrar el equilibrio.

Debido a la forma en la que se ha planteado la estructura de este proyecto, se decidió usar el motor basado en herencia, ya que la mayor parte de clases que se han pensado en implementar tienen gran parte de funcionalidad en común y propósitos muy similares. Es por eso por lo que en varias de las clases que heredan de otras de la herramienta sobrescribimos métodos de las clases primitivas, ofreciéndonos una gran flexibilidad y muchas posibilidades para entrar en detalles concretos en alguna clase que requiera de más funcionalidad.

5.3 Arquitectura

Para el desarrollo de la aplicación se realizó en primera instancia un prototipo funcional sin un claro modelo arquitectónico, pero que sirvió de gran ayuda para tener las ideas claras, y aprender a manejar de forma más rápida e intuitiva la principal biblioteca que serviría para desarrollar la herramienta: ThreeJS. Esta primera iteración también se ha utilizado para definir lo que se quiere llegar a hacer a nivel de funcionalidad.

En la segunda iteración se ha decidido hacer una refactorización de código con el objetivo de definir una arquitectura más extensible y que permita desarrollar otras versiones de la aplicación más fácilmente. Para ello se ha dividido en tres partes:

- **Engine:** la parte base, la que implementa toda la parte de ThreeJS y la estructura principal del proyecto con el objetivo de que la herramienta sea lo menos intrusiva posible en las funcionalidades más básicas del motor en sí.
- **Simulador:** la parte que extiende e implementa aquellos métodos más complejos que van a depender exclusivamente de las intenciones y objetivos de la aplicación. A su vez está generalizado para que, a través del archivo de datos, se puedan cambiar los atributos que se necesitan usar como referencia para representar correctamente los modelos.
- **Data:** la parte donde se encuentran todos los datos necesarios para implementar tanto la parte base como la parte extendida, desde cada una de las texturas y modelos, hasta cada una de las informaciones más específicas de cada uno de los elementos que componen la aplicación.

Todo esto se puede observar más detalladamente en el diagrama de clases representado en la Figura 5-3.

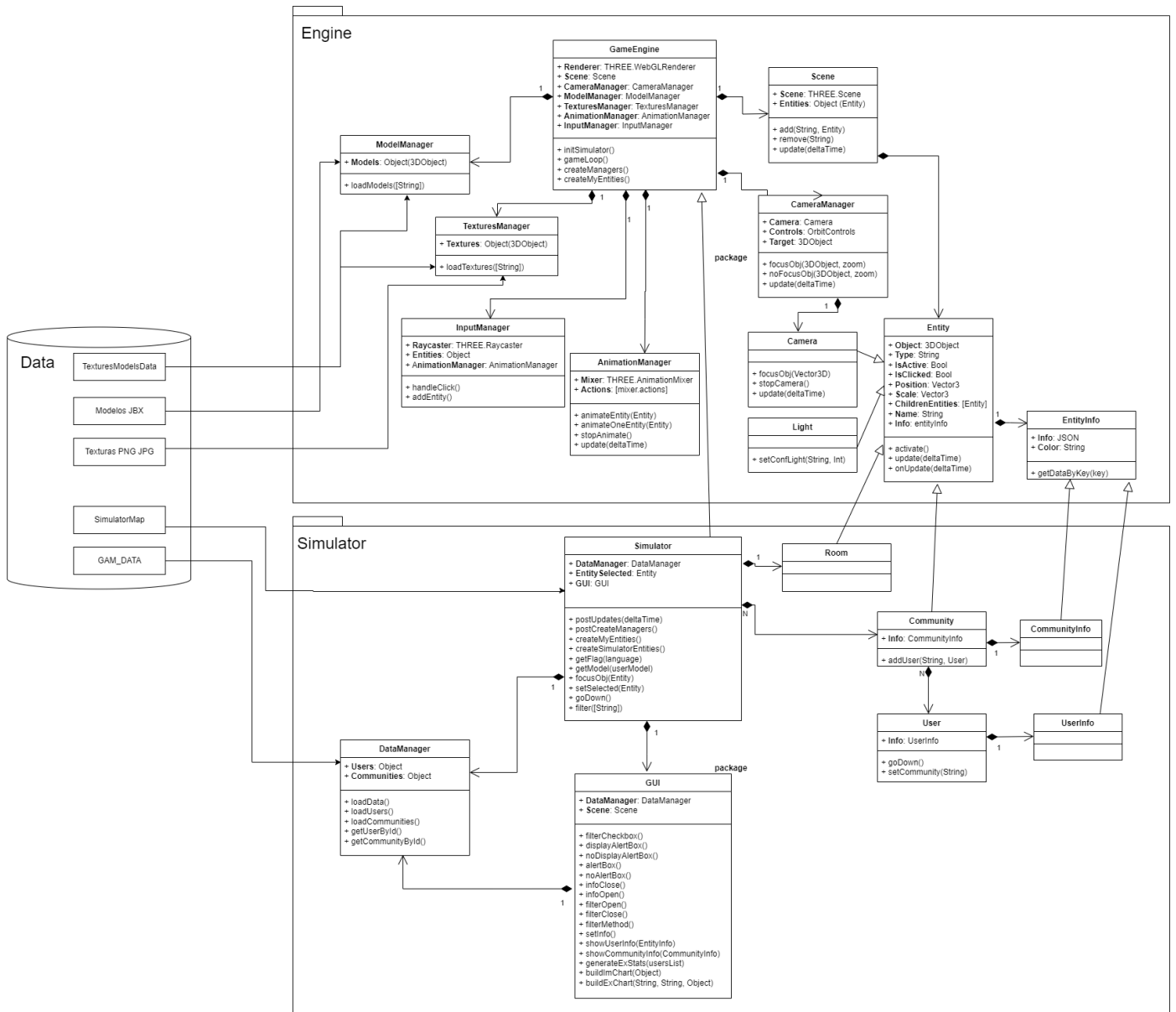


Figura 5-3 Diagrama de clases

5.3.1 Engine

Es la parte central de la aplicación, el corazón de esta. Es la encargada de dar una funcionalidad básica a todo el motor de juego y contiene la principal estructura de funcionamiento para una aplicación web con ThreeJS.

- **GameEngine:** es el motor principal de juego que genera el bucle principal gameLoop y crea la base de ThreeJS con su cámara, escena y render.
- **CameraManager:** es el manager encargado de manejar la cámara y permitir la rotación, desplazamiento y enfocado de la cámara.
- **ModelManager** y **TextureManager:** son los managers encargados de cargar y gestionar todas las texturas y modelos para su posterior uso.
- **InputManager:** se encarga de recibir los diferentes inputs y gestionarlos para mandar los eventos correspondientes al observador, para que este lo gestione posteriormente.
- **AnimationManager:** es la encargada de animar a todas las entidades que contienen animaciones, que han sido previamente cargados en el **ModelManager**.
- **Scene:** es la clase encargada de gestionar todos los elementos representables, añadirlos, eliminarlos y representarlos en su posición correspondiente. Además, actualiza cada uno de ellos en cada iteración del bucle principal generado por **GameEngine**.
- **Entity:** es la clase principal de la cual heredarán de ella todas las unidades representables en 3 dimensiones en nuestro juego. Esta clase se ha realizado para que todas aquellas entidades que tienen una serie de métodos y atributos en común se gestionen de mejor manera y faciliten el acceso y trato. Dos de las entidades básicas para la representación son:
 - **Camera,** es la encargada de dar el punto de vista desde el cual se observa la escena.

- **Light**, es la que proporciona la iluminación adecuada y necesaria para que esta pueda verse de forma correcta.
- **EntityInfo**, es la clase encargada de dar y gestionar información en cada una de las distintas entidades que lo necesiten.

5.3.2 Simulator

Esta parte, es la cual se encarga de ampliar la base del motor de juego y darle toda la funcionalidad y usabilidad específica de nuestro caso de estudio (la visualización de comunidades de ciudadanos). El simulador está implementado de tal forma que , los dos primeros atributos que se especifiquen en el archivo de datos compondrán el modelo del ciudadano, y el tercer atributo será el utilizado para representar los billboard. De esta forma, si se quisieran representar otros atributos en 3D, tan solo habría que sustituir sus nombres y valores en el archivo de datos, y proporcionar la representación adecuada para cada uno en las carpetas correspondientes.

- **Simulator**: Es la clase que hereda de **GameEngine**, es la que amplía y añade nuevos managers y métodos a nuestro motor de juego. Es la encargada de gestionar todo lo relacionado con la simulación de la aplicación.
- **Room**: es la clase encargada de representar la sala principal de la aplicación, donde se distribuyen todas las comunidades y ciudadanos que queremos representar en nuestra aplicación.
- **User**: Es una de las entidades más importantes de la aplicación ya que es la que representa un ciudadano.
 - **UserInfo**: es la clase que hereda de EntityInfo y se encarga de mostrar la información de ciudadano al que corresponde.
- **Community**: Es otra de las entidades más importantes de la aplicación ya que es la que representa una comunidad de ciudadanos.
 - **CommunityInfo**: es la clase que hereda de EntityInfo y es la que se encarga de dar la información de cada comunidad.

- **DataManager:** es la clase encargada de gestionar la información de los ciudadanos y comunidades, darle la forma correspondiente, y almacenarla en los EntityInfo.
- **GUI:** la clase encargada de gestionar y mostrar la parte de la información para el usuario. Es la encargada de mostrar la información de aquella entidad seleccionada. Su funcionamiento consiste en generar un desplegable en la parte superior derecha de la pantalla, en la que se muestra toda la información de la entidad contenida en su **EntityInfo**. También tiene la funcionalidad de desplegar un filtro de ciudadanos en la parte inferior derecha.

5.3.3 Data

Esta es una de las partes más importantes de la aplicación ya que es a través de la cual, toda la parte de la simulación cobra sentido, ya que tenemos varios archivos en formato JSON en los que se encuentra la información de cada uno de los ciudadanos y comunidades, los modelos y texturas a cargar y los diferentes tamaños y posiciones de los demás elementos de la aplicación. Además de contener también los diferentes archivos de imagen o modelado 3D.

- **SimulatorMap:** el encargado de tener los datos de la distribución de los elementos generales de la escena. En él se encuentra el tamaño de la habitación, la posición inicial y general de la cámara y las diferentes distribuciones de ciudadanos y comunidades en la escena.
- **TexturesModelsData:** contiene las referencias a las texturas y modelos que deseamos cargar en nuestra aplicación con su correspondiente clave para luego utilizarlos desde la parte del simulador.
- **Modelos y Texturas:** son diferentes archivos, los modelos creados a través de archivos **fbx**. Y texturas a través de archivos **png** y **jpg**. Todos ellos organizados y cargados desde el **ModelManager** y **TextureManager** y referenciados en el **TexturesModelsData**

- **DATA:** archivo que contiene los diferentes atributos de cada ciudadano a representar, y con ello la información indispensable para la realización de los filtros y cargado de modelos. Además, toda la información de las comunidades, tanto su información interna como la información de cada ciudadano que la compone. Este archivo es específico del proyecto europeo *SPICE*, que es similar al que usar VISIR en sus representaciones. En este archivo se encuentran los diferentes atributos explícitos que se desean representar. En el caso de este Trabajo de Fin de Grado los que componen el modelo, como ya se ha mencionado previamente son la edad y el género, mientras que el atributo que será representado a través del billboard es la lengua. Si se quisieran añadir más atributos, simplemente habría que añadirlos en este archivo, y especificar en el simulador su representación, ya sea una nueva o sirva este atributo para representar una dimensión más a través del modelo tridimensional, por ejemplo. En este último caso, el modelo estaría definido por 3 atributos y habría que incluir todas las combinaciones posibles de los 3 atributos para la representación.

Capítulo 6 - Evaluaciones

6.1 Propósito y objetivos

La evaluación de usuarios es un proceso fundamental en el diseño y desarrollo de aplicaciones. Consiste en recopilar información directamente de los usuarios para comprender cómo interactúan con la aplicación, qué desafíos enfrentan y qué mejoras se pueden implementar. Mediante la realización de pruebas, cuestionarios y observación de los usuarios en acción, se obtiene una valiosa retroalimentación que permite identificar problemas de usabilidad, descubrir oportunidades de optimización y validar la eficacia de las funcionalidades.

A través el proceso de evaluación con usuarios, se busca evaluar y medir la experiencia del usuario para asegurarse de que la información mostrada sea entendible y comprensible. Se busca determinar si el contenido y la presentación de la información se adecuan a las necesidades y expectativas de los usuarios, permitiendo una comprensión clara y efectiva.

Además, otro objetivo clave de la evaluación con usuarios es evaluar si las interacciones con la herramienta son intuitivas. Esto implica analizar si las acciones requeridas por parte del usuario son fáciles de aprender y realizar, evitando la necesidad de instrucciones complejas o esfuerzo adicional para lograr los objetivos deseados.

En última instancia, la evaluación con usuarios busca mejorar la usabilidad y la experiencia del usuario, proporcionando una retroalimentación valiosa para optimizar el diseño y garantizar que se cumplan los propósitos y objetivos establecidos.

6.2 Preguntas de investigación

Para especificar mejor el propósito y los objetivos de la evaluación, se han especificado una serie de preguntas de investigación que deberán ser respondidas tras acabar la evaluación, y que nos ayudarán a comprobar si el prototipo cumple con lo esperado y detectar posibles problemas y fallos.

- ¿El usuario sabe navegar correctamente por los apartados de la aplicación?

- ¿El usuario entiende para qué sirve cada apartado de la aplicación?
- ¿Sabe encontrar la información que necesita el usuario?
- ¿La aplicación es intuitiva?

6.3 Requisitos de los participantes

Los participantes deberán ser personas que tengan relación o hayan tenido relación con el proyecto SPICE, ya que de esta forma nos aseguramos de que toda la información que se muestra es entendida por el participante.

6.4 Metodología

Para cada evaluación se les proporcionará una serie de instrucciones (indicadas en el apartado 6.5.1), para poder manejar la herramienta de forma más cómoda y fluida, y una serie de tareas a realizar con las diferentes utilidades de la herramienta(indicadas en el apartado 6.5.2).

Se van a realizar una serie de evaluaciones de forma online y se irán recopilando los datos y comentarios que la persona evaluada vaya haciendo. Posteriormente se le pasará un formulario SUS que se realizará con Google Forms de forma que todas las respuestas queden unificadas y conseguir más facilidad a la hora de analizar.

Se va a realizar un formulario SUS [6] (System Usability Scale) debido a su facilidad y fiabilidad. Además, incluye una fórmula en base a las respuestas de los usuarios para calcular de forma numérica la usabilidad de la aplicación, con el que podremos sacar los datos estadísticos y la usabilidad de la herramienta.

El formulario SUS que se va a realizar, constará de las siguientes 10 preguntas, evaluadas numéricamente por valores entre 1-5, donde el usuario indica su nivel de acuerdo o desacuerdo con cada afirmación. El formulario SUS para realizar la evaluación se encuentra en el siguiente enlace:

<https://forms.gle/S9d8aa4PfcLi4tw8>

Además, se les proporcionó en el propio formulario una pregunta en la que podían hacer comentarios extra, para tenerlos en cuenta y sacar más información de ellos.

6.5 Material de las evaluaciones con usuarios

Se realizaron 3 evaluaciones a través de meet en las que se fueron encontrando con diferentes problemas, estos datos se han ido recopilando y con ellos rellorando una serie de tablas para realizar mejor el análisis. También se enviaron las tareas a otros usuarios para que las realizaran y rellorasen los cuestionarios, pero al no haber estado presentes, tenemos menos información sobre cómo fueron, salvo el cuestionario SUS que fue común a todos.

6.5.1 Instrucciones

1. Movimiento:

- a. Clic izquierdo + movimiento de ratón: Poder desplazarte rotando sobre el centro de la escena, o de una entidad seleccionada.
- b. Rueda del ratón: Para poder hacer o quitar zoom.
- c. Clic derecho: Poder cancelar el desplazamiento automático de la cámara, generado al pulsar sobre una entidad.

2. Filtrado:

- a. Activar el filtrado: Se activará la parte del checkbox y te permitirá ver los usuarios con ese atributo. Por defecto todos activados.
- b. Desactivar el filtrado: Se desactivará la parte del checkbox y te reducirá la opacidad de los usuarios que tuvieran ese atributo.
- c. Cerrar filtros: Al hacer clic sobre la cruz que hay en el mismo desplegable de los filtros.

3. Información:

- a. Doble clic: Al hacer doble clic y hacerlo encima de una comunidad a usuario, se desplegará un cuadro de información.

- b. Cerrar información y deseleccionar: Al hacer clic sobre la cruz que se encuentra en la parte de información, se cerrará la información y se deseleccionará a la entidad seleccionada, volviendo a la vista general.

6.5.2 Lista de tareas

1. Explorar el entorno 3D, saber desplazar la cámara y poder ver mejor las comunidades y ciudadanos, realizar giros de cámara y zoom.
2. Seleccionar una comunidad y revisar sus datos, sus estadísticas.
3. Cerrar su información, y volver al plano general.
4. Seleccionar un ciudadano y revisar sus datos y los datos de la comunidad a la que pertenece, y su obra más relevante.
5. Cerrar toda la información del usuario y volver al plano general.
6. Aplicar filtros a los ciudadanos y ver su funcionamiento

6.6 Resultados evaluaciones

En este capítulo se exponen los datos recopilados en las tres sesiones de evaluación que se llevaron a cabo con tres personas familiarizadas tanto con el proyecto SPICE como con la herramienta VISIR en dos dimensiones.

6.6.1.1 Evaluación 1

Tarea	Problema	Comentario
2. Seleccionar una comunidad y revisar sus datos, sus estadísticas	No entiendo por qué los colores distintos dentro de la comunidad.	Nos comenta que, si no tienen una utilidad concreta, es mejor quitarlo, porque genera mucho ruido.

3. Cerrar su información, y volver al plano general.	No se deseleccionar una comunidad a simple vista	Al leer posteriormente en la tarea, "y cerrar la información" ya reconoció la forma de hacerlo, haciendo clic en la cruz y elaboró la tarea. Nos proporcionó la idea de poder hacer otros métodos para deseleccionar la información.
4. Seleccionar un ciudadano y revisar sus datos y los datos de la comunidad a la que pertenece, y su obra más relevante.		Nos sugirió si podíamos dar más información sobre la obra de arte más relevante del usuario, como si dijo algo, o sintió alguna emoción hacia ella

Tabla 11 Datos de evaluación 1

6.6.1.2 Evaluación 2

Tarea	Problema	Comentario
1. Explorar el entorno 3D, saber desplazar la cámara y poder ver mejor las comunidades y ciudadanos, realizar giros de cámara y zoom.		La persona evaluada piensa que el efecto de la cámara de volver al plano general es un efecto indeseado
3.Cerrar la información de una comunidad, y volver al plano general.	Trata de hacer clic fuera de la comunidad	La primera idea de la persona evaluada para cerrar el cuadro informativo es hacer clic fuera de la comunidad, en vez de sobre la cruz superior derecha.

Tabla 22 Datos de evaluación 2

6.6.1.3 Evaluación 3

Tarea	Problema	Comentario
2. Seleccionar una comunidad y revisar sus datos, sus estadísticas	Hace clic en vez de doble clic	Ligera confusión para la persona que estaba realizando la evaluación ya que de forma intuitiva hace solo un clic.
3. Cerrar la información de una comunidad, y volver al plano general.	Trata de cerrar el cuadro informativo pulsando fuera de la comunidad y con la tecla de escape	Aquí el usuario hace las dos cosas que se le vienen primero a la cabeza para cerrar el cuadro de información, aunque en cuestión de un rápido vistazo lo corrige y cierra bien el cuadro.

Tabla 33 Datos de Evaluación 3

6.6.2 Otras evaluaciones

También se realizaron otras evaluaciones, en las que no estuvimos presentes. Estos usuarios cumplen los requisitos establecidos en el apartado 6.3. Gracias a estas otras evaluaciones tenemos más y mejores resultados sobre la herramienta. A estos usuarios se les proporcionó la lista de tareas junto con las instrucciones para poder realizar la evaluación. Y finalmente el formulario SUS con la pregunta de comentarios, en la que se le dio especial importancia, ya que es donde nos podían proporcionar una retroalimentación de sus opiniones, problemas y sugerencias.

6.7 Análisis de resultados

En este apartado se van a analizar las evaluaciones tomando como datos los contenidos en las tablas anteriores, y se van a realizar una tabla más elaborada para sacar las conclusiones. De igual forma se va a realizar un análisis del formulario SUS que se ha pasado a cada participante.

6.7.1 Tabla de análisis

En la Tabla 4 muestra un análisis más concreto de los problemas que se han presentado en las evaluaciones, y a su vez se proporciona una solución con un coste, dado entre los valores 1-5 siendo 1 muy costoso y 5 poco costoso, y una severidad, dado entre los valores 1-5 siendo 1 poco severo y 5 muy severo, para posteriormente calcular la prioridad de solución, ésta se calcula multiplicando los valores del coste y de la severidad, y asignando valor en función de los tramos, es decir, si el cálculo de la multiplicación se encuentra entre el rango 1-5 entonces tendrá prioridad 1, si es 6-10 será prioridad 2, y así hasta llegar a la máxima prioridad, la 5, siendo el rango 21-25, la prioridad se toma en cuenta, siendo 1 poco prioritario, y 5 muy prioritario.

Id	Descripción del problema	Solución	Coste	Severidad	Prioridad
1.2	No encuentra lógica en encontrarse diferentes colores al seleccionar las diferentes comunidades.	Asignar un único color a todas las comunidades, porque es verdad que solo añade ruido al no tener un significado propio.	5	1	1
1.3 2.3 3.3	Cerrar la información y deseleccionar una comunidad dándole solo a la cruz .	Añadir funcionalidad para que también se pueda deseleccionar y cerrar la información haciendo doble clic sobre cualquier otra parte del plano en la que no interfiera ninguna entidad.	3	3	2
3.2	Hacer un clic para seleccionar en vez de hacer doble clic.	Especificar de forma más visual las instrucciones de uso de la aplicación.	5	2	2

Tabla 4 Tabla análisis final

6.7.2 Formulario SUS

El formulario SUS nos va a servir para determinar si nuestra herramienta tiene una buena usabilidad. Este formulario se ha convertido en un estándar en la industria principalmente porque ofrece una escala muy fácil de administrar a los participantes y porque se puede utilizar con muestras muy pequeñas para conseguir resultados fiables. Esta última razón es uno de los motivos esenciales por el que hemos decidido realizarlo, ya que nuestros participantes son muy limitados, ya que tienen que ser expertos en la materia.

6.7.2.1 Resultados

Siguiendo las instrucciones de uso del cuestionario SUS, la media obtenida en el formulario por nuestra aplicación en cada pregunta es la siguiente:

Pregunta	Respuesta	Cálculo	Contribución
1. I think that I would like to use this system frequently.	3.75	3.75-1	2.75
2. I found the system unnecessarily complex.	1.25	1.25-5	3.75
3. I thought the system was easy to use.	4.75	4.75-1	3.75
4. I think that I would need the support of a technical person to be able to use this system.	1	1-5	4
5. I found the various functions in this system were well integrated.	4.25	4.25-1	3.25
6. I thought there was too much inconsistency in this system.	1.25	1.25-5	3.75
7. I would imagine that most people would learn to use this system very quickly.	4.5	4.5-1	3.5
8. I found the system very cumbersome to use.	1	1-5	4
9. I felt very confident using the system.	4.5	4.5-1	3.5
10. I needed to learn a lot of things before I could get going with this system.	2.75	2.75-5	2.25

Tabla 55 Respuestas formulario SUS

Dado que la suma de las contribuciones es 34.5, al multiplicarla por 2.5 el resultado obtenido es de 86.25. Según los criterios estándares de los formularios SUS, cualquier resultado por encima de 68 se considera por encima de la media, por lo que la puntuación de nuestra herramienta se encuentra en ese sector. El análisis del formulario SUS se realizó con las respuestas recopiladas hasta el día 27 de mayo. De todas formas, el formulario continuará abierto para que en las futuras evaluaciones que se hagan se pueda sacar más información y conseguir mejores resultados.

6.7.3 Otros comentarios

Como se ha comentado en el apartado 6.4 se ha proporcionado a los usuarios en la parte final del formulario SUS un cuadro de texto en el que se les animaba a dejar sus opiniones y sugerencias sobre la herramienta, para tener información más precisa y exacta. Aquí hay dos de los comentarios que fueron recopilados:

"For curators with limited skills and knowledge in computer science (or in general in concepts like clustering, emotions detection etc..), I suggest including a more explanatory legend to clarify the different percentages of extracted emotions with different properties of the cluster respect with the single selected person in a single community."

"The system seems to be intuitive and well functioning - takes a little bit of time to learn to navigate without a mouse but generally, great work!"

6.7.4 Conclusiones

Tras haber realizado un estudio de las diferentes evaluaciones que se realizaron, podemos ver que los resultados son buenos, apenas se encontraron problemas en las tareas, y los pocos que se encontraron tenían una severidad baja, además de tener una sencilla solución, que no costó ser implementada.

Basándonos en los resultados del formulario SUS podemos extraer que los participantes expresaron interés moderado en utilizar el sistema con frecuencia. En general indican que el sistema es fácil de usar y no necesitarían de una persona especializada para saber utilizarlo correctamente.

También extraemos que las diferentes funciones que tiene la herramienta están bien integradas entre sí. Se sintieron cómodos al utilizarla y opinan que las personas que quieran usarlo no van a tener problemas en aprender a utilizarlo en la mayoría de los casos. Sin embargo, algunos usuarios respondieron que encontraban cierta complejidad y que necesitarían aprender bien las instrucciones antes de usarlo, por lo que en un futuro podría ser un aspecto que mejorar.

En resumen, los resultados del formulario han sido satisfactorios según las personas evaluadas, e indica que la aplicación tiene un uso fácil, aunque pueden existir algunos problemas sin el aprendizaje inicial relacionados con este aspecto.

Capítulo 7 - Conclusiones y trabajo futuro

7.1 Revisión de objetivos

Al inicio del desarrollo de la herramienta se propusieron una serie de objetivos, que podemos encontrar en el apartado 1.2, como esenciales para llevar a cabo el proyecto de forma correcta.

- El primero consistía en analizar la visualización de las redes en tres dimensiones y las herramientas y librerías que aportaría más valor al trabajo realizado. Dicho análisis se realizó de forma correcta y recopilando información que, gracias a ella, se ha utilizado para desarrollar la herramienta correctamente.
- También se propuso realizar un estudio sobre que atributos de las comunidades de ciudadanos tienen sentido ser representados en 3D y cuál sería la forma más apropiada de modelarlos. En este mismo estudio se planteó determinar cómo poder mostrar la información no modelable en 3D. Llevando a cabo el estudio se encontraron los atributos más representativos en 3D y con ello se generaron los modelos. De igual forma, en el caso de los no modelables, se generó una interfaz donde se muestran para que el usuario pueda acceder a ellos sin problema y no pierda la información.
- Por último, se planteó generar un primer prototipo de la herramienta para posteriormente desarrollar un motor generalizado que pudiera ser usado como prototipo de herramienta generalizado, y de esta forma permitir ser reusado de cara a otros proyectos futuros. Este planteamiento ha tenido muy buenos resultados, ya que en el primer prototipo se consiguió mucha experiencia en el entorno de desarrollo y gracias a ellos el segundo, en el que se implementó en motor generalizado, ha sido mucho más potente y ofrece la posibilidad de reusarlo de cara a otros proyectos.

7.2 Trabajo futuro

Existen ciertos aspectos que han surgido durante el desarrollo del trabajo que no han podido ser implementados o estudiados de cara a la versión final del proyecto, pero que sería interesante explorar más adelante.

En primer lugar, una de las funcionalidades más atractivas de VISIR, que permitía observar dos perspectivas al mismo tiempo ofrece un punto de vista muy eficaz a la hora de comparar dos tipos de medidas de similitud diferentes. Esto es por la comodidad que ofrece al usuario la capacidad de ver a los mismos ciudadanos organizados de forma diferente por dos algoritmos de detección de comunidades distintos en paralelo.

Es por esto por lo que en un principio se propuso la idea de implementar la posibilidad de disponer de dos habitaciones en el viewport al mismo tiempo. Cada una de las habitaciones, claro está, representaría a una de las perspectivas. De tal modo que, si se seleccionase un ciudadano en una de las habitaciones, se marcara como seleccionado también en la otra habitación para facilitar la comparación del resultado de la colocación de dicho ciudadano.

7.3 Conclusión

En general, el desarrollo e implementación de la herramienta ha tenido un resultado bueno, cumpliendo todos los objetivos propuestos al inicio de este. Además, gracias a las evaluaciones realizadas y a sus conclusiones (apartado 6.7.4), podemos confirmar que la herramienta tiene un acabo final agradable y con una alta usabilidad. En estas mismas evaluaciones se detectaron una serie de problemas que ya han sido corregidos.

Además, podemos encontrar un gran número de mejoras a implementar, ya que esta herramienta puede tener un gran potencial de cara al objetivo principal de la herramienta, la visualización en 3D de las comunidades de ciudadanos. Una de las mejoras posibles que se podría realizar en cuanto a las tres dimensiones, es el hecho de aprovechar mucho más la tercera dimensión, ya que, en nuestro caso, utilizamos el plano con los ejes x e y para colocar a los diferentes ciudadanos de las comunidades, y el eje z para elevar los modelos cuando son seleccionados y poder observar sus

atributos de mejor forma. Además de utilizar las tres dimensiones para dar mucho más detalle a los ciudadanos y con ello ofrecer la posibilidad de mostrar más atributos. Por ello creemos que se le puede dar mucho más nivel de uso a la dimensión en el eje z.

Capítulo 8 - Conclusions and future work

8.1 Objective Review

At the beginning of the tool's development, a series of goals were proposed, which can be found in section 2.2, as essential to carry out the project correctly.

- The first objective was to analyze the visualization of networks in three dimensions and the tools and libraries that would contribute the most value to the project. This analysis was successfully conducted, providing valuable information that was used to develop the tool correctly.
- It was also proposed to study which attributes of citizen communities make sense to be represented in 3D and what would be the most appropriate way to model them. This study also aimed to determine how to display non-modelable information in 3D. Through this study, the most representative attributes in 3D were identified, and models were generated accordingly. Additionally, for non-modelable attributes, an interface was created to display them so that users can access the information without any issues.
- Lastly, the idea was to generate an initial prototype of the tool and subsequently develop a generalized engine that could be used as a prototype for other future projects. This approach has yielded excellent results, as the first prototype provided valuable development experience, enabling the creation of a more powerful second prototype with a generalized engine that can be reused for other projects.

8.2 Future work

There are certain aspects that have emerged during the development of the project that could not be implemented or studied for the final version but would be interesting to explore in the future.

Firstly, one of the most appealing features of VISIR, which allowed observing two perspectives simultaneously, offers an effective viewpoint when comparing different types of similarity measures. This is due to the convenience it provides to the user by being able to see the same individuals organized differently by two different community detection algorithms in parallel.

For this reason, the idea was initially proposed to implement the possibility of having two rooms in the viewport at the same time. Each room would represent one of the perspectives. Therefore, if a citizen were selected in one room, they would also be marked as selected in the other room to facilitate the comparison of the result of placing that citizen.

8.3 Conclusion

Overall, the development and implementation of the tool have yielded positive results, fulfilling all the objectives set at the beginning. Furthermore, based on the evaluations conducted and their conclusions (section 6.7.4), we can confirm that the tool has a pleasant final outcome and high usability. Several identified issues from these evaluations have already been addressed.

Moreover, there are numerous potential improvements that can be made since this tool has significant potential for its primary objective of visualizing citizen communities in 3D. One possible improvement regarding the three dimensions is to make better use of the third dimension. In our case, we use the XY plane to position citizens of different communities, and the Z-axis to elevate models when selected for better attribute observation. Expanding the utilization of the three dimensions would provide more detail to citizens and offer the opportunity to display additional attributes. Therefore, we believe that there is much more potential to explore in the Z-axis dimension.

CONTRIBUCIONES PERSONALES

David Lago Hernández

Esta aplicación se ha gestionado en dos iteraciones, una primera en la que hicimos una idea general, pero sin una estructura clara y limpia. Y una segunda iteración donde se hizo la gran parte de ella en conjunto. Es decir, se fue quedando de forma presencial u online y se fueron realizando avances en la aplicación.

En la primera iteración yo me encargué de la realización más relacionada con la herramienta ThreeJS, es decir, los siguientes puntos:

- De la realización de la primera escena y renderizado, junto a sus primeras luces, estudiando previamente cada una de las que nos proporciona ThreeJS y sus diferentes estilos.
- De la realización de pruebas con la cámara, desde el movimiento, hasta la colocación y desplazamiento automático al hacer clic sobre las comunidades o usuarios para poder dar una mejor perspectiva y visualización de estos.
- De la exploración más a fondo del maquetado de los modelos y su correcta carga y renderizado y de la propia creación de ellos (trabajo que se hizo en conjunto).
- De la creación de las diferentes entidades y la organización de las diferentes funciones de cada una de ellas dándole así su funcionamiento correcto dentro de la aplicación.
- La distribución en la habitación de las diferentes comunidades a través de las distintas formas geométricas elegidas de forma común, para así colocar de forma correcta tanto a las comunidades en la habitación como a los usuarios dentro de cada comunidad.
- El despliegue en Docker de la aplicación, para facilitar así su utilización desde los diferentes ordenadores en los que se quisiera visualizar.

En la segunda iteración, aunque la mayoría se realizó de forma común, hay cosas que si se realizaron más de forma individual.

- Realización de la estructura principal de la aplicación tomando como referencia un motor de juego basado en ThreeJS.
- La implementación de Webpack para el mejor uso y empaquetado de las diferentes librerías implicadas en la aplicación.
- La creación de la clase principal Entity y sus correctas funciones para ejercer como clase "padre" de todas las demás clases que la heredan.
- La creación de cada uno de los elementos de la aplicación, desde la propia habitación principal en la que se encuentran todas las comunidades y sus correspondientes miembros. Como la propia implementación del funcionamiento de los miembros.
- El funcionamiento interno del motor de juego, con su correspondiente bucle y sus diferentes actualizaciones en cada uno de los elementos que lo necesitaran.
- El estudio de cómo realizar de forma correcta el despliegue de la aplicación en GitHub Pages. Así como comprobar y arreglar todos los pequeños problemas que encontré a la hora de desplegarlo.
- El desarrollo de la parte del *InputManager* relacionada con el pase de datos desde el simulador a la parte visual de la GUI, y con ello organizar la información para mostrarla.
- La reestructuración y modelado correctamente del documento JSON con la información de cada uno de los ciudadanos para su correcta lectura y uso en la aplicación. Esto se debió a que se tenía un prototipo del JSON con menos datos de los que nosotros necesitábamos.

Ignacio Urretavizcaya Tato

De cara a la primera versión implementada mi trabajo se centró, principalmente, en construir la interacción del usuario con la herramienta, tanto a nivel de interfaz gráfica como de interacción con el entorno 3D. También puse foco en una primera versión de los modelos en tres dimensiones. Esto se puede resumir en:

- Implementación de la interfaz gráfica y las estructuras sobre las que se muestra información, tanto como el desarrollo e implementación de las animaciones y estilos de la interfaz para tratar de enriquecer la experiencia de uso.
- Implementación de la interacción del usuario con la interfaz gráfica, controlando la lógica detrás de cada acción en función del caso en el que la interacción sucedía.
- Implementación de la interacción con las entidades mostradas en tres dimensiones,
- Desarrollo de la lógica para controlar las distintas combinaciones de acciones del usuario con los diferentes modelos representados en 3D, como los ciudadanos y las comunidades.
- Acoplamiento de la interacción con los modelos de tres dimensiones con la interfaz gráfica para que fueran en conjunto y que la interfaz respondiera a las diferentes acciones posibles relacionadas con las entidades.
- Investigación y modelaje de las primeras entidades en tres dimensiones, así como la forma de importarlas a la herramienta.
- Implementación de la entidad de cámara, como de su movimiento y transiciones entre distintas posiciones.

Por otro lado, En cuanto a la segunda iteración trabajé continuamente junto a mi compañero en prácticamente la totalidad del desarrollo, lo que incluye personalmente la implementación de las diferentes clases principales de las que heredan las demás, como el nuevo modelo de interacción y la funcionalidad de los filtros y en el apartado de la representación de información en la interfaz gráfica:

- Diseño de la interfaz gráfica del cuadro de filtrado, y el trabajo de generalización para el posible intercambio de atributos en la simulación.
- Diseño de la lógica del filtrado y el funcionamiento profundo de dicha funcionalidad en consecuencia del estado del cuadro de filtrado.
- Estudio de diferentes opciones a la hora de representar los atributos estadísticos de las comunidades y ciudadanos, valorando opciones como D3.js y Chart.js, la que finalmente fue escogida
- Implementación y uso de Chart.js para mostrar la información explícita e implícita de una comunidad.
- Desarrollo tanto de la clase principal *Entity*, como de sus métodos generales que servirán como apoyo para las clases que hereden de ella.
- El desarrollo de la clase *InputManager*, rediseñando por completo el modelo de interacción del usuario con la aplicación y el tratamiento de eventos entre el usuario y la interfaz o entidades.
- Desarrollo de los distintos modelos tridimensionales que componen la escena en su totalidad, desde la habitación, hasta los modelos usados para los ciudadanos, pasando también por las comunidades y su representación gráfica.

BIBLIOGRAFÍA

- [1] Á. González, «Force-Directed Graph UOC,» [En línea]. Available: <http://personal.uoc.edu/VIS/D3/ES/force-directed/index.html>.
- [2] «Webpack,» [En línea]. Available: <https://webpack.js.org/>.
- [3] «D3.js,» [En línea]. Available: <https://www.d3js.org/>.
- [4] Chartjs. [En línea]. Available: <https://www.chartjs.org/> .
- [5] Game Engine Architecture.
- [6] «System Usability Scale (SUS),» [En línea]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.

