Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

ANALISIS Y VISUALIZACION DE GRAFOS USANDO REALIDAD VIRTUAL: UNA PERSPECTIVA DE INTERACCION

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **ING. ALEJANDRO DANIEL ORTEGA ROSALES**

Asesor **DR. LUIS FERNANDO GUTIERREZ PRECIADO**

Tlaquepaque, Jalisco. enero de 2020.

AGRADECIMIENTOS

Primeramente, agradezco al Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) por otorgarme un descuento en el costo total del plan de estudios. Agradezco específicamente a la Maestría en Sistemas Computacionales por permitirme crecer académica y profesionalmente al ser parte de su cuerpo estudiantil.

También quiero agradecer a mi madre, Luz Elena, así como a mis hermanos Mayra y Arturo, por su sostén y consejo durante mi desarrollo académico y personal.

Agradezco a mi pareja Andrea Tejada por su apoyo, consejo y compañía durante el caminar de esta aventura académica y más allá de la misma.

Quiero agradecer a todos los profesores que compartieron sus conocimientos conmigo, especialmente a la Dr. Luis Fernando Gutierrez Preciado, por su constante guía durante mi proceso de aprendizaje sobre la investigación científica en esta maestría.

Esta tesis se ha podido desarrollar adecuadamente a partir del apoyo otorgado por el Consejo Nacional de Ciencia y Tecnología (CONACYT) por medio de la beca Nacional con número 487260 la cual se me fue otorgada durante el periodo que comprendió la Maestría en Sistemas Computacionales del ITESO.

Agradezco a INTEL, la compañía en la que actualmente trabajo, por apoyarme a cubrir el total de los gatos remanentes a los apoyos anteriormente mencionados.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE ACRÓNIMOS Y ABREVIATURAS

TOG	Trabajo de Obtención de Grado
VR	Virtual Reality
HMD	Head Mounted Display
API	Application Programing Interface
SQL	Structured Query Language
CAVE	Cave Automatic Virtual Enviroment
LCD	Liquid Crystal Display
2D	Two Dimensions
3D	Three Dimensions
MEMS	Microelectromechanical systems
SDK	Software Development Kit
CAVE	Cave Assisted Virtual Environment
LCD	Liquid Cristal Display
SCI	Scientific Computing and Imaging Institute
RDF	Resource Description Framework
MR	Mixed Reality

1. RESUMEN

Actualmente el análisis de grafos se realiza principalmente en ambientes bidimensionales mediante la aplicación y el uso de métodos o herramientas limitadas a usuarios expertos en teoría de grafos. Así mismo, un grafo masivo es extremadamente difícil de diseñar en 2D (dos dimensiones) de una manera que ayude a las personas a entenderlo con facilidad. Se propone una solución genérica para la visualización y análisis de grafos en ambientes virtuales 3D utilizando como interfaz de usuario el dispositivo HMD (*Head Mount Display*) de la marca HTC Vive.

Este proyecto no está programado para trabajar con un conjunto de datos o formato en particular, se propone una arquitectura novedosa capaz de visualizar cualquier tipo de dato que pueda ser manipulado por NEO4J [1]. Se discutirá en mayor medida las aportaciones hechas para la parte de interacción e interfaz de usuario con el ambiente virtual.

Durante el desarrollo del proyecto fue necesario realizar modificaciones a los componentes de Unity [2] para optimizar el renderizado del grafo y no afectar el desempeño del CPU.

2. INTRODUCCIÓN

La teoría de grafos está fundamentada en las matemáticas discretas, requiere conocimientos de diversas áreas de estudio como el álgebra, aritmética, geometría de polígonos, probabilidad y lógica combinatoria. Actualmente es aplicada principalmente en el campo de las ciencias computacionales, telecomunicaciones e informática.

Se han desarrollado aplicaciones matemáticas para analizar grafos cuyos resultados pueden ser difíciles de interpretar, por lo que visualizar grafos se ha convertido en una herramienta común para que los usuarios obtengan una comprensión intuitiva de los datos. Si bien la investigación sobre algoritmos de visualización de grafos es extensa, gran parte de esta investigación se ha centrado en la visualización bidimensional que a menudo produce estructuras densas y confusas "tipo espagueti" que eluden el análisis visual y la comprensión [3].

La teoría de grafos es usada para resolver diversos problemas en distintas áreas de estudios. Resolución de algoritmos, administración de proyectos, síntesis de circuitos secuenciales, dibujo computacional, modelado de trayectorias de comunicación, análisis de redes sociales, control de producción, entre otros. Este proyecto podrá ser utilizado por estudiantes e investigadores interesados en el análisis y visualización de redes de información sin conocimientos previos de teoría de grafos dado que la navegación, visualización y análisis se realizará de manera intuitiva.

2.1. Antecedentes

El problema de representar visualmente grandes cantidades de datos ha sido abordado por los investigadores mediante plataformas de análisis y visión limitado a 2D como la aplicación Gephi [4]. Las investigaciones enfocadas a la interacción y representación visual de grafos en 3D (tres dimensiones) utilizado dispositivos VR (*Virtual Reality*) son escasas y están ligadas a un ambiente educacional o limitadas a un tipo redes en específico como el proyecto iCAVE (*interactome*-CAVE) [5].

2.2. Justificación

Recientemente ha habido un aumento en la popularidad de los dispositivos VR [6]. Estos dispositivos permiten la posibilidad de inmersión en entornos de visualización estereoscópicos. Las técnicas actuales que utilizan tales ambientes inmersivos han sido exploradas extensivamente para visualizaciones científicas y espaciales, contrastantemente muy poco se ha explorado para la visualización de la información basada en grafos.

2.3. Problema

La visualización se ha convertido en un medio establecido para ayudar a las personas en varios campos a comprender mejor la información. Sin embargo, a medida que el tamaño de los datos aumenta, los enfoques de visualización tienen que lidiar con el problema de representaciones visuales abarrotadas y superpobladas de información. Además, la complejidad de los datos hace que sea difícil codificar toda la información relevante de un conjunto de datos en una sola imagen de visualización bidimensional. Existe la necesidad de contar con una interfaz de usuario intuitiva para el análisis de grafos en VR con el objetivo de reducir la curva de aprendizaje a nuevos usuarios.

2.4. Mercado meta

El trabajo está dirigido a cualquier grupo y/o individuo interesado en el análisis y representación visual de información masiva utilizando herramientas interactivas. El conocimiento específico en aplicaciones de software o teoría de grafos no será una limitante, por lo que personas del área de Psicología, Sociología y Antropología tendrán acceso relativamente sencillo e inmediato al análisis de redes sociales mediante grafos. La herramienta podrá ser utilizada de igual manera por personas interesadas en aplicaciones mercadológicas y políticas.

2.5. Características funcionales

La herramienta funcionará exclusivamente para el dispositivo HTC Vive utilizando la plataforma de desarrollo Unity configurada para aplicaciones realidad virtual. El proyecto está dividido en tres secciones principales: la interfaz de usuario en realidad virtual, visualización de grafos en 3D y análisis de grafos. El enfoque de este proyecto en particular es para la primera sección, interfaz de usuario en realidad virtual. Se deberá de capturar e interpretar la interacción del usuario con el dispositivo para el manejo del grafo mediante las opciones de visualización, interacción y filtros proporcionados por la interfaz, mismas que serán codificadas en función de los avances de las dos secciones del proyecto restantes.

Se programo en lenguaje C# debido a la facilidad de programación en la plataforma Unity para el desarrollo de aplicaciones VR. El proyecto no está pensado para ser comercializado, pero existe la posibilidad de utilizar la herramienta de visualización y análisis de grafos para programar aplicaciones

móviles con fines educativos. Se busca hacer más eficiente el acceso a la información en repositorios de datos masivos y facilitar su análisis.

3. ESTADO DEL ARTE O DE LA TÉCNICA

Resumen: En este capítulo se presenta un resumen de los trabajos relacionados con el uso de la tecnología de realidad virtual para la interacción con grafos.

3.1. iCAVE

La herramienta de código abierto iCAVE fue desarrollada en 2017 para la visualización de redes biomoleculares en 3D [5]. Es completamente portable y funciona con la mayoría de las tarjetas gráficas comerciales. Ofrece la posibilidad de visualizar redes masivas, complejas y multicapa en tres distintas configuraciones con el fin de adaptarse a los recursos tecnológicos del usuario. Se puede lograr la visualización e interacción en 3D con un monitor estándar, ratón y teclado. Ofrece la posibilidad de configurar un ambiente estereoscópico utilizando unas gafas 3D y un monitor capaz de desplegar imágenes estereoscópicas. Idealmente iCAVE se utiliza en instalaciones CAVE (*Cave Assisted Virtual Environment*) en donde el usuario puede interactuar en un ambiente de inmersión con el grafo.

Dentro de la interfaz de usuario de iCAVE, los investigadores pueden alternar fácilmente entre algoritmos de diseño para enfatizar diferentes aspectos de la red. Los usuarios que utilicen ratón como dispositivo de interacción pueden rotar, hacer acercamientos o escalar el grafo para investigar estructuras de interés dentro de la red, mandar a imprimir capturas de pantalla y guardar video al momento de rotar el grafo. La rotación con mouse permite el análisis de múltiples puntos de vista para usuarios que no cuenten con equipo 3D.

Aquellos que cuenten con una instalación CAVE podrán hacer uso de la interacción de inmersión utilizando unas gafas de obturación LCD (*Liquid Cristal Display*) estereoscópicas que transmiten imágenes en 3D. Cuando el usuario camina, un arreglo de sensores rastrea la posición del ojo del usuario y ajusta la perspectiva de visualización en función de los movimientos del usuario. La interacción es realizada mediante el uso de una varita de proyección de rayos para realizar movimientos que son mapeados a eventos lógicos manejados por la aplicación de diseño de la red. El acercamiento y rotación son activados con la misma varita apuntando a un nodo o una arista en particular. Múltiples usuarios pueden existir simultáneamente en la red y visualizar desde múltiples perspectivas moviéndose en el

espacio o interactuando directamente con biomoléculas específicas haciéndoles clic con la varita para mostrar todas sus interacciones en esa red y almacenarlas en una base de datos.

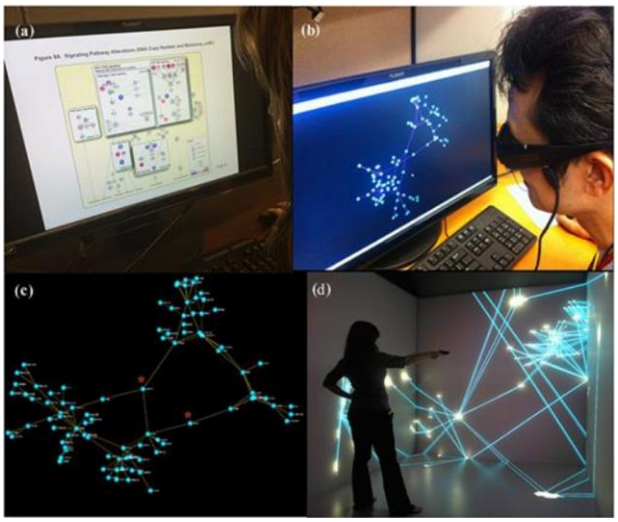

Figura 3.1 Diferentes configuraciones de visualización e interacción utilizando iCAVE [5]

iCAVE hace uso de la tecnología VR pensado para la visualización y análisis de redes biomoleculares. Los algoritmos de visualización y filtros que implementa están enfocados a este tipo de redes. Para explotar todas sus capacidades se necesita contar con una instalación CAVE la cual es costosa ya que incluye un gran número de proyectores y sensores distribuidos en una cuarto cubico para crear la sensación de estar dentro de un mundo virtual. No es compatible con Windows ya que las librerías utilizadas en el proyecto no permiten dicha compatibilidad.

3.2. Virtual Reality Visualization Tool for Neuron Tracing

Los investigadores Will Usher y Pavol Klacansky del instituto SCI (*Scientific Computing and Imaging Institute*) de la universidad de Utah, USA. Desarrollaron una herramienta de visualización de redes neuronales utilizando tecnología comercial VR en 2017 [7]. Proporcionan una herramienta accesible de bajo costo comparada con una instalación CAVE, es escalable y se enfoca en el rastreo de neuronas y sus conexiones espaciales en conjuntos de datos de gran magnitud comparados a los que se pueden procesar actualmente con herramientas 2D.

El primer prototipo fue programado utilizando el dispositivo VR Oculus Rift HMD [8]. El Oculus Rift puede rastrear pequeños movimientos de la cabeza, pero no admite capturar información al caminar y no se puede utilizar para alcanzar o "tocar" los datos directamente debido a las limitantes de su mando. El deseo de manipulación de información en 3D los llevó a buscar un paradigma de interacción diferente. Con ese fin, se trasladaron a la plataforma HTC Vive [6], que admite VR a escala de habitación e incluye controladores de seguimiento. El seguimiento a escala de habitación permite a los usuarios caminar e interactuar con la red de forma natural utilizando sus manos, similar a una instalación CAVE.

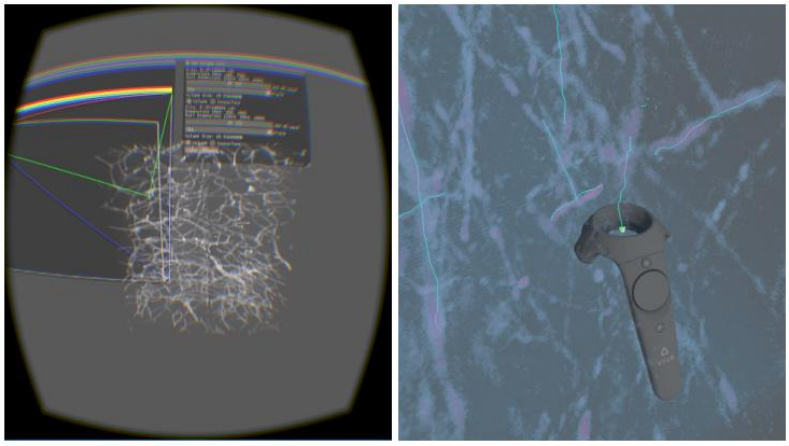

Figura 3.2 Visualización científica y rastreo de neuronas en realidad virtual [7]

La herramienta propuesta está ligada a la investigación en neurociencias, la mayoría de sus funciones y algoritmos están optimizados para poder realizar tareas similares a las utilizadas en NeuroLucida [9], el estándar actual de la industria para el rastreo de neuronas en una red.

3.3. Redgraph

Redgraph es una herramienta basada en realidad virtual para el análisis de datos de la web semántica. Los usuarios comienzan visualizando el grafo en 2D y mediante interacción extraen la parte del grafo de interés a un ambiente en 3D [3].

Figura 3.3 Usuarios de Redgraph interactuando en sala VR.

El proyecto está enfocado a procesar datos que puedan ser descritos mediante el modelo RDF (*Resource Description Framework*) para facilitar el mapeo del grafo y con la limitante de que los datos tienen que leerse desde un archivo en formato GraphXML.

Una de sus principales ventajas es que puede utilizar los algoritmos de distribución existentes para 2D y conseguir un mapeo inicial del grafo adecuado. También cuenta con una extensión que permite modificar un algoritmo de distribución existente para mapear el grafo en 3D.

Su mayor limitante es como ellos mencionan la forma en que van a colocar al usuario en el ambiente 3D ya que una vez colocado el usuario, no se cuenta con la función de desplazamiento, solo con la interacción de la red. Otra limitante es que el proyecto está enfocado a un tipo y formato de datos en particular por lo que se requiere bastante preprocesamiento de datos antes de mandarlos al ambiente en VR.

3.4. Microsoft HoloLens 3D network visualizer

El Proyecto utiliza los HoloLens [10] de Microsoft los cuales utilizan tecnología MR (*Mixed Reality*) para mapear redes de computadoras en un ambiente 3D con el objetivo de monitorear la red de una manera más intuitiva [11].

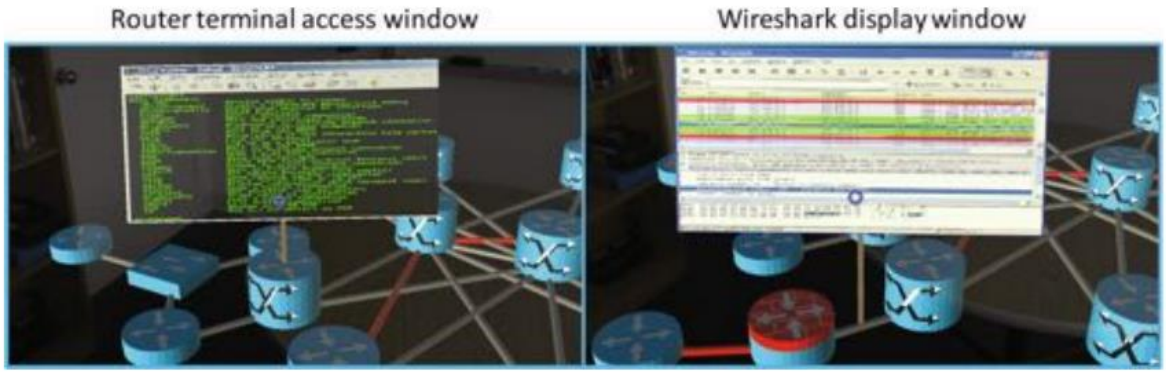

Figura 3.4 Interactuando con una red virtual utilizando HoloLens

HoloLens es un dispositivo que utiliza gestos físicos del usuario como interfaz con el mundo virtual. Gran parte del proyecto consistió en programar un software en Windows Mixed Reality [12] que permite compatibilidad con el dispositivo. Posteriormente se portó la aplicación que desarrollaron a la plataforma Unity para hacer uso del HoloToolkit.

Este proyecto en particular tiene bastantes limitantes, principalmente por tener que hacer uso de las aplicaciones y dispositivos de Microsoft y únicamente trabaja con archivos de formato GraphML. El objetivo del proyecto se inclina más a demostrar las capacidades del dispositivo HoloLens que en lograr realmente un análisis e interacción adecuados con una red en 3D.

4. MARCO TEÓRICO/CONCEPTUAL

Resumen: En este capítulo se presentan las bases teóricas y conceptuales sobre la implementación y programación de dispositivos VR HMD para lograr una interacción intuitiva con un grafo proyectado en un ambiente 3D.

4.1. Unity

Unity es un motor de videojuego multiplataforma creado por Unity Technologies [2]. Unity está disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas como el HTC Vive para realidad virtual.

El éxito de Unity ha llegado en parte debido al enfoque en las necesidades de los desarrolladores independientes que no pueden crear ni su propio motor del juego ni las herramientas necesarias o adquirir licencias para utilizar plenamente las opciones que aparecen disponibles. El enfoque de la compañía es "democratizar el desarrollo de juegos", y hacer el desarrollo de contenidos interactivos en 2D y 3D lo más accesible posible a tantas personas en todo el mundo como sea posible [2].

4.2. HTC Vive

Vive son unas gafas de realidad virtual fabricadas por HTC y Valve. El dispositivo está diseñado para utilizar el espacio en una habitación y sumergirse en un mundo virtual en el que se permite al usuario caminar y utilizar controladores para interactuar con objetos virtuales [6].

HTC afirma que él Vive tiene una frecuencia de actualización de 90 Hz. El dispositivo utiliza dos pantallas, una para cada ojo, cada una con una resolución de 1080x1200. Utiliza más de 70 sensores, incluyendo un giroscopio MEMS (*microelectromechanical systems*), acelerómetros y sensores láser, y está hecho para funcionar en un área se seguimiento de 4.6 metros por 4.6 metros, teniendo una precisión de menos de un milímetro. El sistema de seguimiento, llamado *Lighthouse* (controles o mandos de interface), fue diseñado por Alan Yates y utiliza foto-sensores para el seguimiento de los objetos; para evitar problemas de oclusión él Vive combina dos *Lighthouse* que barren todo un espacio con láseres de luz estructurada. La cámara frontal permite detectar cualquier objeto, estático o en movimiento, en un

área; Está función sirve también como sistema de seguridad, mostrando el mundo real para evitar que los usuarios choquen con objetos.

4.3. VRTK

VRTK (*Virtual Reality Toolkit*) es una colección de scripts y conceptos útiles para ayudar a construir soluciones de realidad virtual rápida y fácilmente en Unity3D [12]. Cubre una serie de soluciones comunes, tales como:

- Locomoción dentro del espacio virtual.
- Interacciones como tocar, agarrar y usar objetos.
- Interactuando con los elementos de la interfaz de usuario de Unity3d a través de punteros.
- Física del cuerpo dentro del espacio virtual.
- Controles 2D y 3D como botones, palancas, puertas y cajones.

4.4. SteamVR SDK

SteamVR es un SDK (*Software Development Kit*) de código abierto que permite a los desarrolladores enfocarse en una única interfaz de controladores que funciona con los principales dispositivos de realidad virtual, incluyendo las experiencias de configuración a escala de habitación que se pueden lograr utilizando el HTC Vive [13]. Además, proporciona acceso a controladores de rastreo y modelos de representación para dispositivos interacción en función del hardware que se esté usando.

SteamVR SDK se importa como una extensión a los proyectos de Unity 3D. Lamentablemente los desarrolladores no proporcionan ningún tipo de documentación bajo la justificación de que es de código abierto. El código es complicado de entender, cuenta con varias capas de abstracción, un gran número de librerías, objetos generados en tiempo de ejecución y es sumamente genérico debido a que está pensado para soportar múltiples dispositivos VR.

Una de las principales ventajas es la posibilidad de simular en Unity 3D un dispositivo de realidad virtual e interactuar utilizando ratón y teclado sin la necesidad de adquirir y configurar un dispositivito VR.

5. ARQUITECTURA DE LA SOLUCION

Existen dos tipos de implementaciones para lograr el funcionamiento de un objeto en Unity. Por un lado, tenemos la definición de la clase con sus miembros y métodos. Por otro lado, tenemos el componente de Unity asociado a dicha clase. Los componentes son objetos que serán renderizados en el espacio VR y tendrán acceso a todos los métodos de su clase, dichos métodos son usados para cambiar color, tamaño, peso, entre otras propiedades.

5.1. Diagrama de componentes.

En la figura 5.1 se muestra el diagrama de componentes que conforman la arquitectura del proyecto. Se muestran ordenados de acuerdo con el flujo que sigue la aplicación para interpretar y renderizar el grafo de entrada.

GEXF File: Es el archivo que contiene al grafo. Este archivo es procesado por la base de datos NEO4j para posteriormente crear una estructura basada en los componentes del grafo.

Parseo: Se crean los componentes del grafo, nodos y aristas con dirección.

Nodos: Es el componente más básico del grafo y es definido por la base de datos, ejemplo, un tweet o un usuario. Cuenta con múltiples métodos para poder ser manipulado y leído por cualquier componente de Unity.

Aristas: La arista representa la relación entre los nodos. Tiene dirección y cuenta con prácticamente los mismos métodos de la clase nodo. Al ser un componente también podemos cambiar su grosor, color y transparencia.

Grafo: Este componente es el más importante de la arquitectura ya que cuenta con listas para nodos y aristas necesarias para manipular los componentes de manera generalizada, por ejemplo, eliminar visualmente los nodos y aristas que no estén dentro de la selección del usuario. Por medio de este componente todos los demás componentes de la arquitectura tienen acceso a manipular nodos, aristas y el grafo en general.

API de visualización: La API de visualización es una serie de métodos embebidos en las clases nodo, arista y grafo, los cuales son utilizados para cambiar la apariencia de los componentes.

API de interacción: Esta API agrega métodos a las clases nodo, arista y grafo para funcionar en conjunto con la API de visualización. Agrega básicamente todos los componentes del proyecto fuera de los componentes básicos del grafo los cuales son necesarios para que el usuario pueda interactuar con el grafo.

VR Toolkit: Este toolkit es la base de la API de interacción, es usado para el renderizado e interacción con los controles del HTC vive. Cada control cuenta con una clase independiente, el control izquierdo está enfocado a la selección de nodos y aristas, mientras que el control derecho está enfocado al menú y paneles de interacción, búsqueda y análisis. En conjunto los dos controles son usados para lograr el desplazamiento en VR.

Controles Cámara: Se utiliza el SDK de SteamVR para lograr la interacción en general del casco con las estaciones base por medio del tracking en los sensores del casco. Este SDK es el encargado de controlar la cámara y está configurado para trabajar en conjunto con el VR toolkit.

Co-routines: Las co-routines son usadas para procesar información en paralelo al renderizado de los componentes, por ejemplo, correr un algoritmo de centralidad sin tener que detener el renderizado.

Medidas de centralidad y algoritmos: Existen medidas de centralidad y algoritmos básicos en el proyecto ya que inicialmente se programó una solución autosuficiente para el análisis de grafos dentro del Unity. En la actualidad los grafos son previamente procesados por la base de datos de Neo4j.

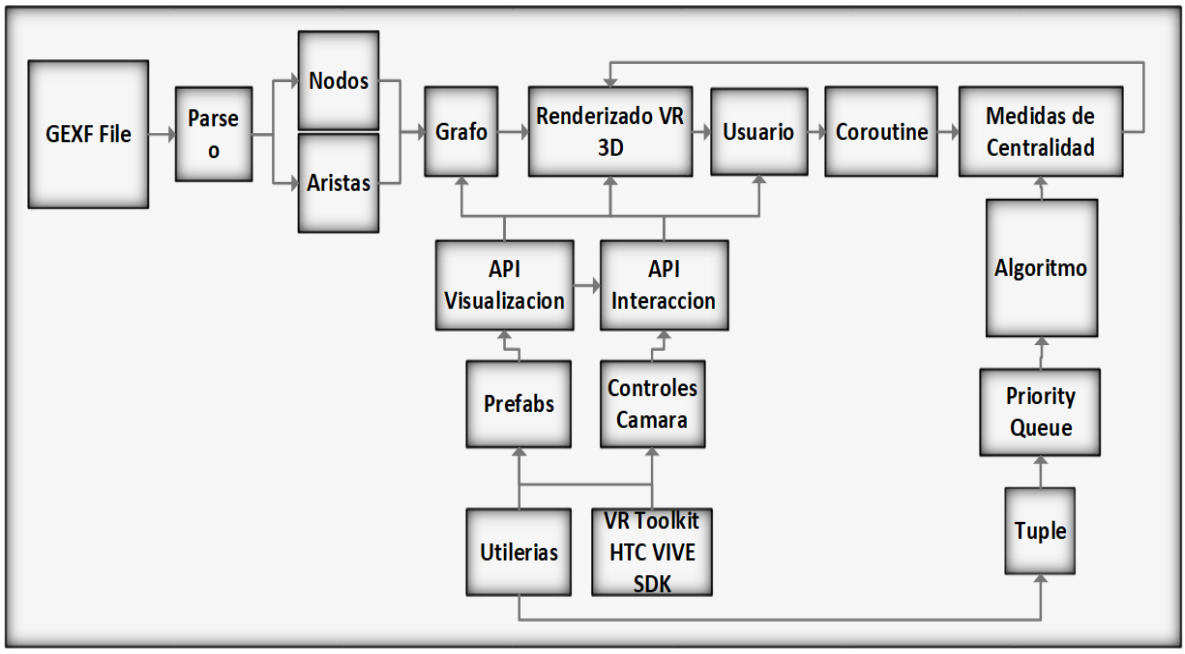

Figura 5.1 Diagrama de componentes

5.2. Diagrama de clases.

La organización del proyecto se muestra en el diagrama de clases de la figura 5.2. La estructura base de la solución está compuesta por las clases nodo, arista y grafo. Adicionalmente se integraron clases auxiliares para el mapeo de archivos XML o base de datos de Neo4j a componentes base para la creación del grafo. La API de visualización describe los componentes base del grafo y su funcionalidad en un contexto de realidad virtual. La API de interacción utiliza la funcionalidad de los componentes ya creados para interactuar con ellos mediante los mandos de usuario.

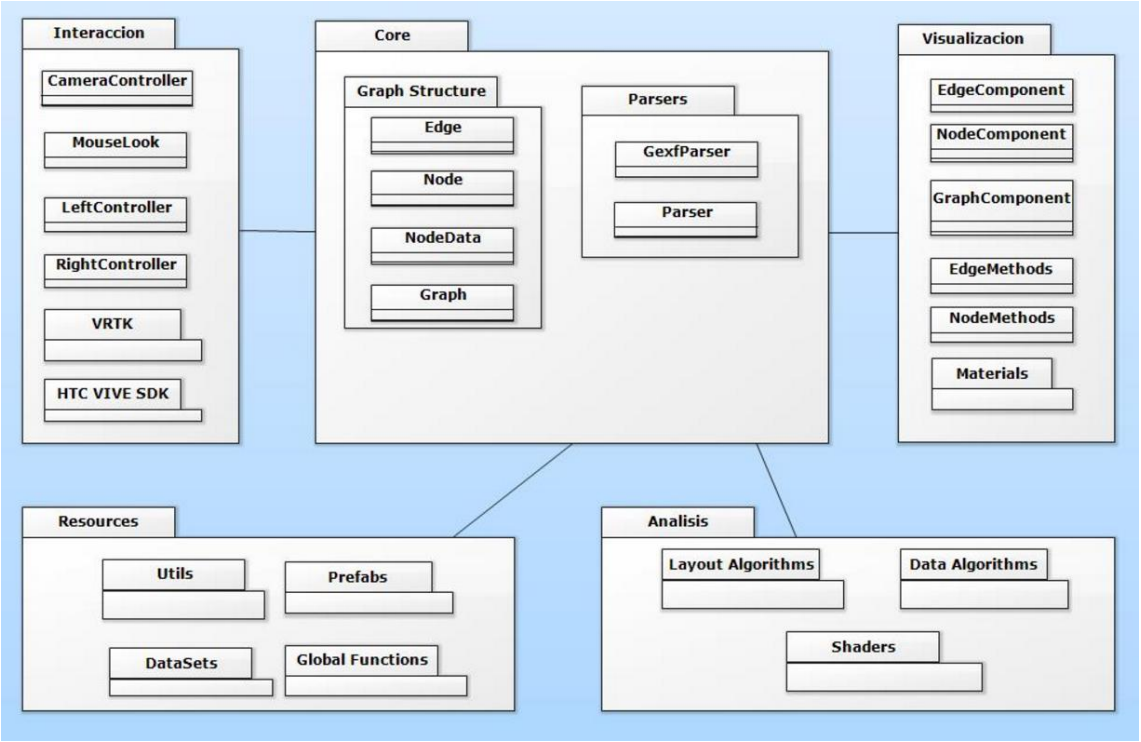

Figura 5.2 Diagrama de clases del proyecto VR.

5.3. Conclusiones

Este capítulo se enfocó en describir el proceso de integración y optimización de las APIs de visualización e interacción mediante el uso del modelo base y de la manera en que fueron implementadas las funcionalidades básicas de interacción.

La arquitectura del proyecto fue pensada para ser programada por tres personas desarrollando cada una de ellas su propia API en paralelo y que en algún punto del desarrollo pudiesen ser integradas con facilidad. La API de interacción se especializó en traducir el grafo de Neo4j a un ambiente virtual eliminando la

complejidad de analizar los datos y crear el grafo dentro de Unity a diferencia de cómo se realizó en el proyecto de Microsoft HoloLens [11].

Se logró concluir con la estructura organizacional y funcionalidad básica del proyecto dando pie a la implementación de funcionalidad avanzada y la integración de algoritmos para el procesamiento de datos que generen información descriptiva de la red que se está analizando.

6. IMPLEMENTACIÓN

6.1. Implementación de la API de interacción.

La API de interacción fue creada en base a un conjunto de herramientas para realidad virtual distribuido gratuitamente bajo el nombre de VRTK. Una caracteriza fundamental de VRTK es que ofrece un simulador VR el cual permite trabajar en cualquier computadora con el uso de ratón y teclado sin la necesidad de contar con un dispositivo de realidad virtual como el HTC VIVE. En la figura 6.1 se muestra el primer grafo renderizado en VR utilizado componentes prefabricados costosos, es decir, componentes nodo/arista que interactuar con una fuente de luz externa direccional, obligando al CPU a calcular por cada componente el sombreado en funciona de la luz incidente.

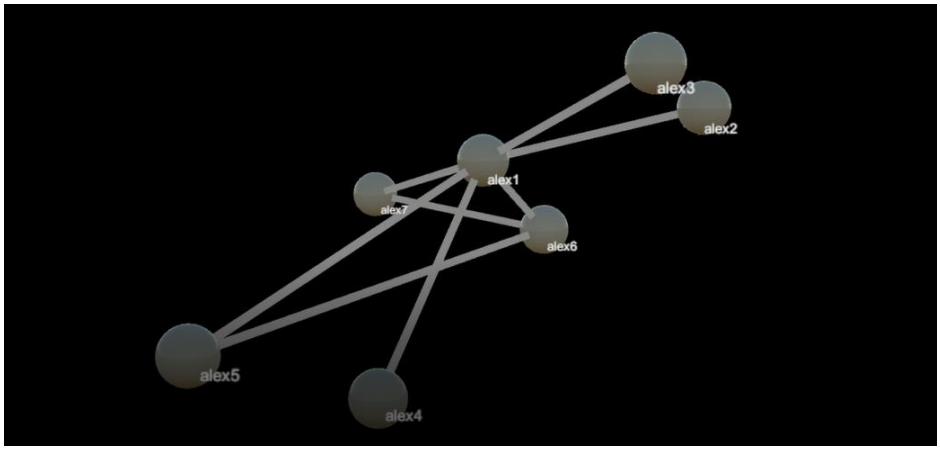

Figura 6.1 Primer grafo en VR utilizando componentes prefabricados de la API de interacción.

Para reducir la carga computacional, se decidió eliminar la luz direccional y cambiar los componentes nodo/arista por componentes basados en *Self-Illumin Shaders* lo cual les permite iluminarse a sí mismos sin necesidad de interactuar con una luz externa. En la figura 6.2 se muestra un grafo con más de 6 mil nodos y 11 mil aristas renderizado con los nuevos componentes sin comprometer el rendimiento del CPU.

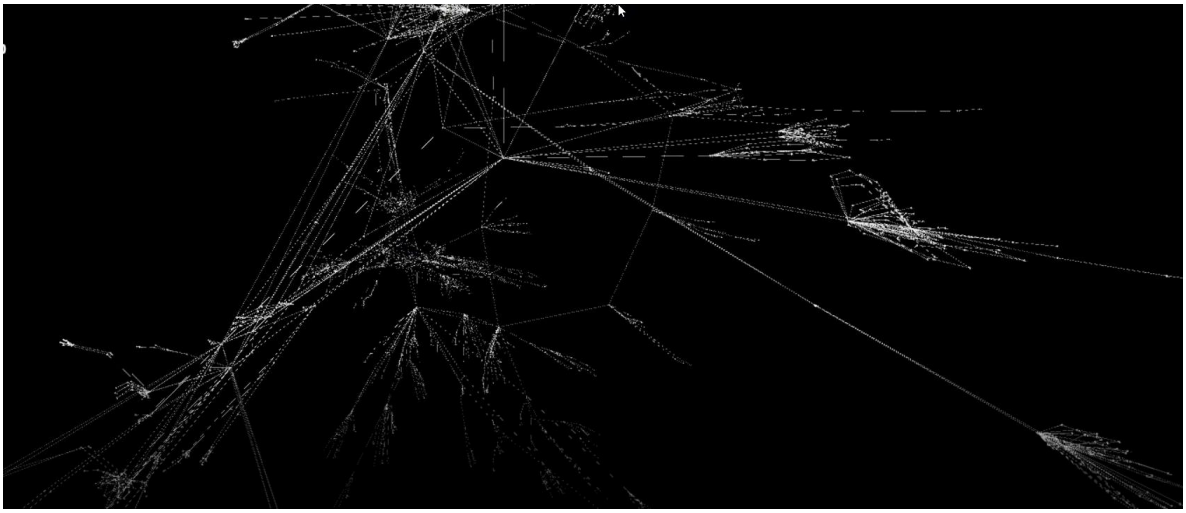

Figura 6.2 Grafo de aproximadamente de 6 mil nodos y 11 aristas en VR.

Utilizando los componentes del VRTK como base, se crearon diversos scripts en C# para el control de la cámara. Uno de los scripts se encarga de modificar la rotación de la cámara de usuario mapeando el movimiento del ratón. Se cuenta con otro script para modificar la posición de la cámara de usuario utilizando el teclado. La figura 6.3 muestra como la cámara de usuario se desplazó en el mundo virtual hasta una región particular del grafo.

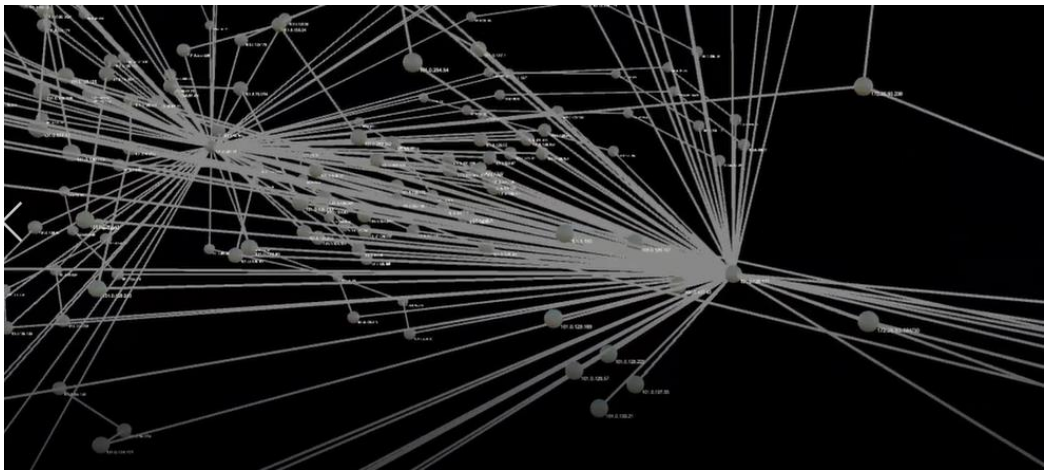

Figura 6.3 Funcionalidad de desplazamiento de la API de interacción.

6.2. Integración de la API de visualización.

La API de visualización propone un modelo base para las clases nodo, arista y grafo. Implementa clases para el mapeo de información desde un archivo XML a componentes del grafo. En la figura 6.4 se muestra el grafo propuesto por la API de visualización con sus componentes base distribuidos aleatoriamente en una esfera de 50 metros de radio. Se hace uso de las funciones de cambio de color para nodos y aristas.

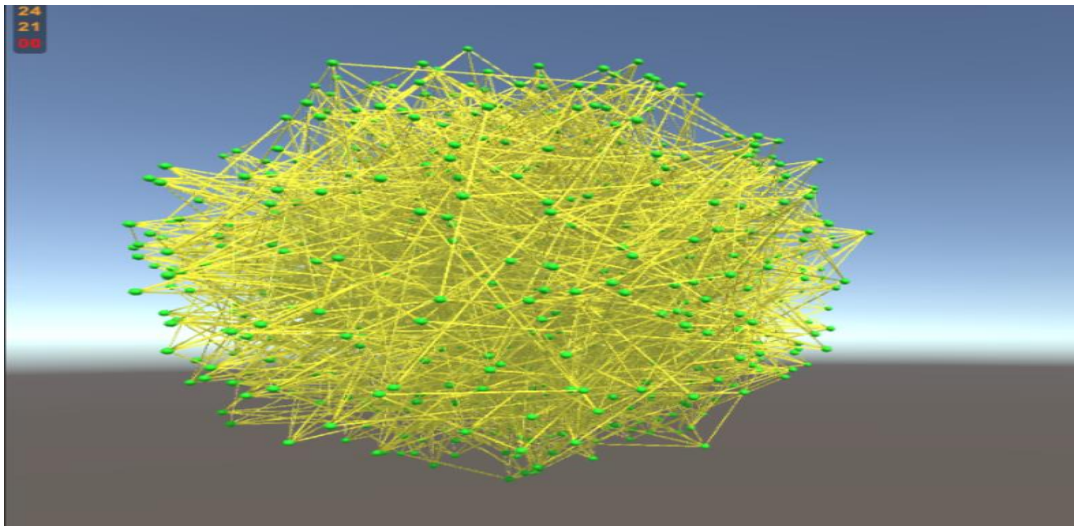

Figura 6.4 Grafo renderizado en VR utilizando componentes prefabricados de la API de visualización.

La integración inicial de las clases de interacción con las clases de visualización se muestra en la figura 6.5. Se puede ver una pobre renderización del grafo dado que el CPU quedo sobre cargado por el procesamiento de los componentes prefabricados de la API de visualización.

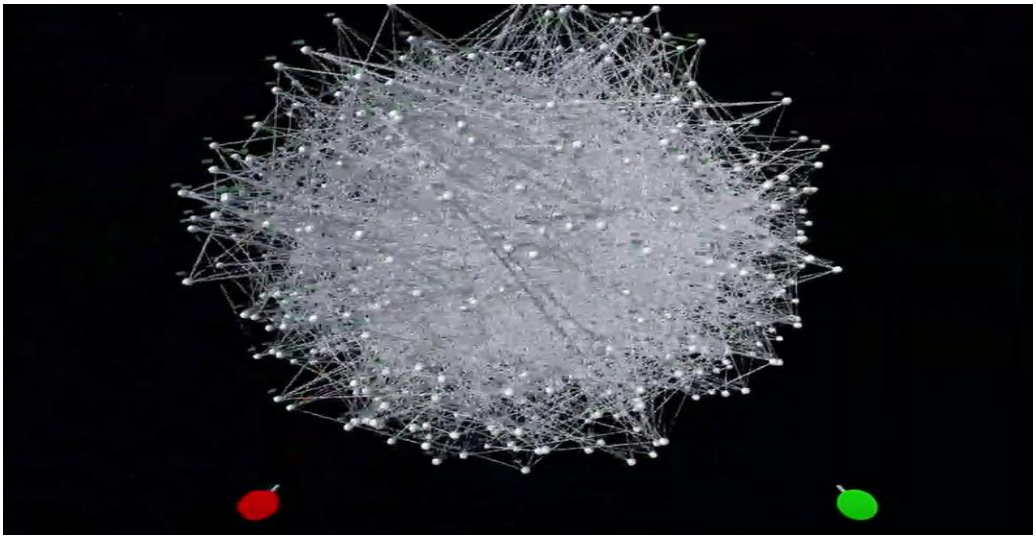

Figura 6.5 Integración de la API de visualización al proyecto de interacción.

Los nodos y aristas son componentes que interactúan con una fuente de luz direccional externa. En la figura 6.6 se puede ver que para las aristas se utilizó un componente del tipo cilindro el cual está compuesto

por una red de 88 vértices y para los nodos un componente tipo esfera compuesto por 515 vértices. La red está compuesta por 824 nodos y 5279 aristas por lo que se están procesando aproximadamente 300 mil primitivas graficas resultando en una gran carga para el CPU, en consecuencia, una pobre renderización como la mostrada en la figura 6.6.

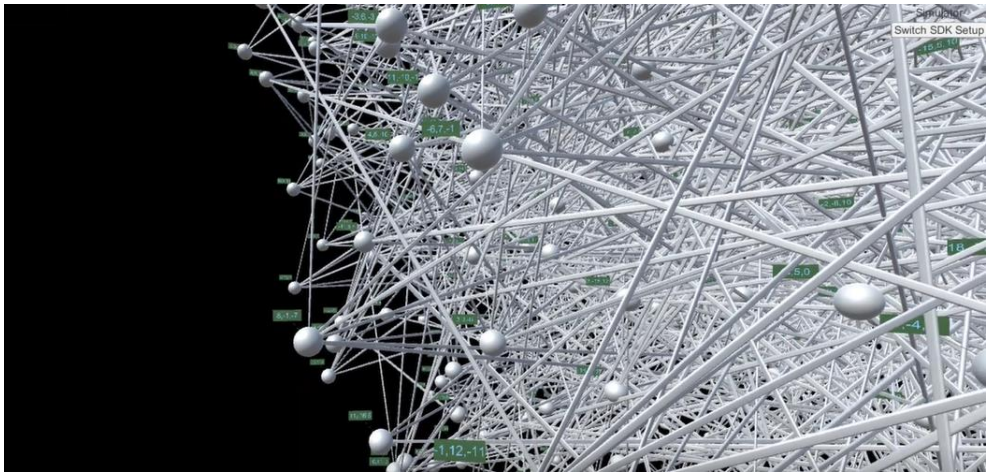

Figura 6.6 Aplicación de desplazamiento y de etiquetado de nodos.

6.3. Optimización de la integración entre APIs.

Para hacer uso óptimo de la funcionalidad de las dos APIs se optó por cambiar los componentes prefabricados del proyecto de interacción conservando la organización de clases propuesta por la API de visualización. Las aristas se remplazaron por cubos compuestos por 24 vértices, podrían estar descritas como un segmento compuesto por 8 vértices, pero esa posibilidad queda descartada dado que *collider* (objeto de Unity usado para representar colisiones a nivel de primitivas) del tipo segmento no funciona adecuadamente para la fácil interacción con el rayo de selección del usuario. En la figura 6.7 se muestra la red en su forma original (sin distribución aleatoria en una esfera) descrita en el archivo XML. Esto se logró al modificar/agregar funciones a la clase de mapeo con el objetivo de conservar la posición original de los elementos del grafo en un supuesto de que una base de datos externa (NEO4J) creara el archivo de topología XML con la posición de los nodos/aristas pre-calculada en función de un algoritmo de *layout* (algoritmo de distribución de nodos en un grafo).

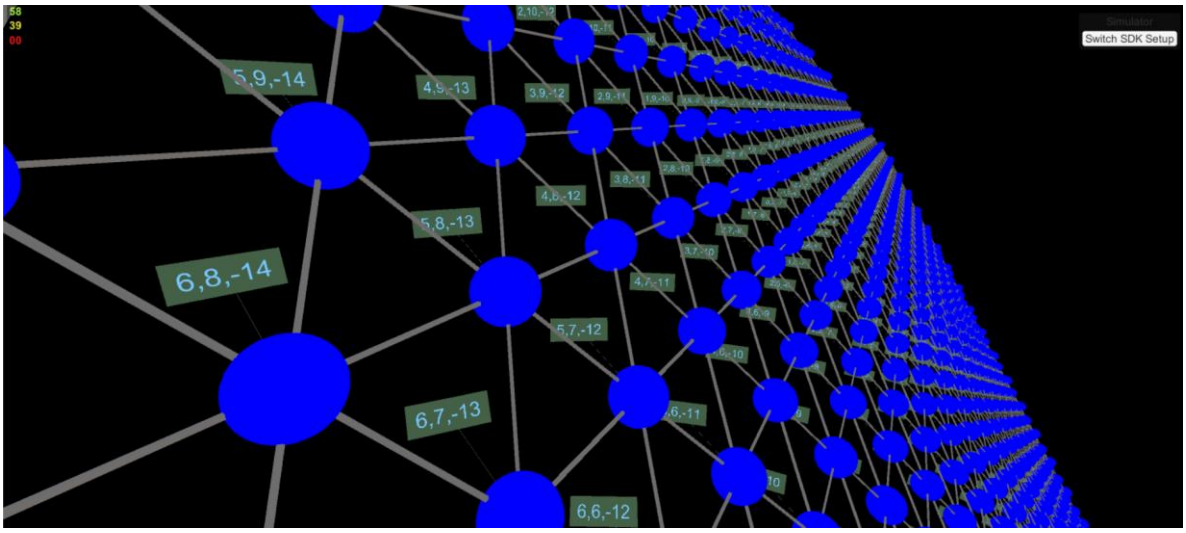

Figura 6.7 Integración de los componentes nodo/arista prefabricados del proyecto de interacción.

En grafos masivos, los nodos serán renderizados usando componentes cúbicos de Unity los cuales están compuestos de 12 primitivas graficas (12 triángulos) para evitar caídas de cuadros por segundos y ralentizado de la aplicación. La clase nodo se modificó para soportar la integración de etiquetas proporcionada por el VRTK. Las etiquetas permiten identificar visualmente a los nodos en tiempo de ejecución y tienen la capacidad de modificar su posición angular para asegurar que siempre apuntaran a la posición angular de la cámara de usuario.

La API de interacción cuenta con dos scripts que describen el comportamiento del mando izquierdo y derecho del usuario. La funcionalidad de cada mando puede ser distinta y estará dada por el script correspondiente. Visualmente el mando derecho se representa como una esfera verde mientras que el mando izquierdo como una esfera roja. Los dos tienen la capacidad de selección mediante rayo.

El mando izquierdo tiene adicionalmente la capacidad de teletransportarse hasta el componente que este seleccionando a una distancia máxima de mil metros. El mando derecho tiene la capacidad de eliminar componentes visualmente del grafo como se muestra en la figura 6.8. Se cuenta con una estructura de datos tipo diccionario en la cual se almacenan los elementos que estén actualmente seleccionados por el usuario con el objetivo de realizar acciones grupales de eliminación, desplazamiento, etiquetado etcétera.

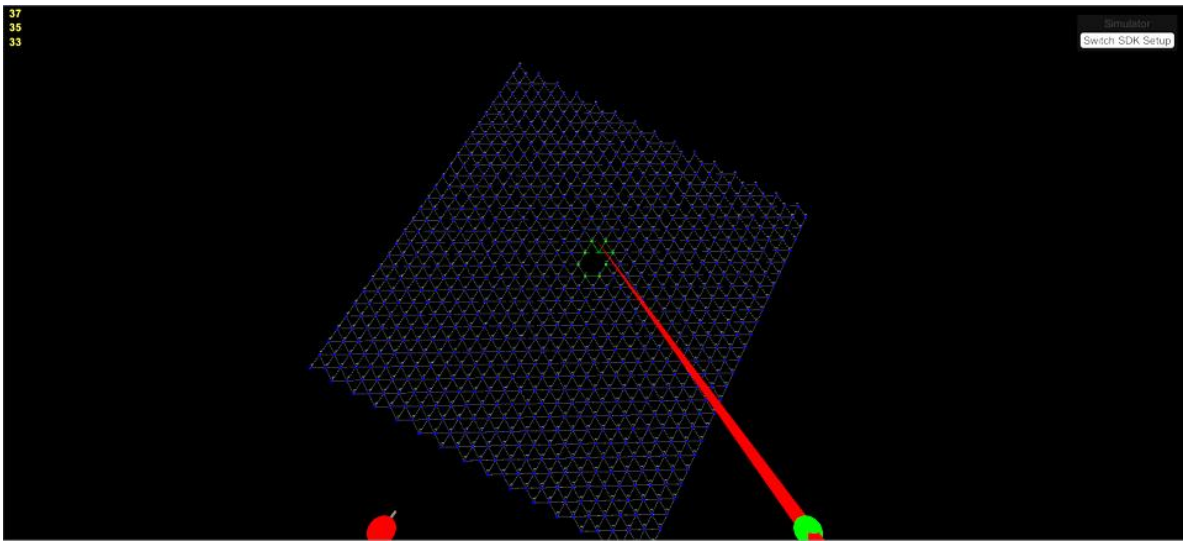

Figura 6.8 Renderizado del grafo en su forma original como red. Se integra la selección y eliminación de componentes.

Los cuadros por segundo a los que corre la aplicación se muestran en la parte superior izquierda de la pantalla. En la figura 6.9 se muestra la renderización de toda la red con los elementos optimizados y las APIs completamente integradas resultando en una ganancia de cuadros por segundo considerable con respecto a las etapas iniciales representadas por las figuras 6.5 y 6.6.

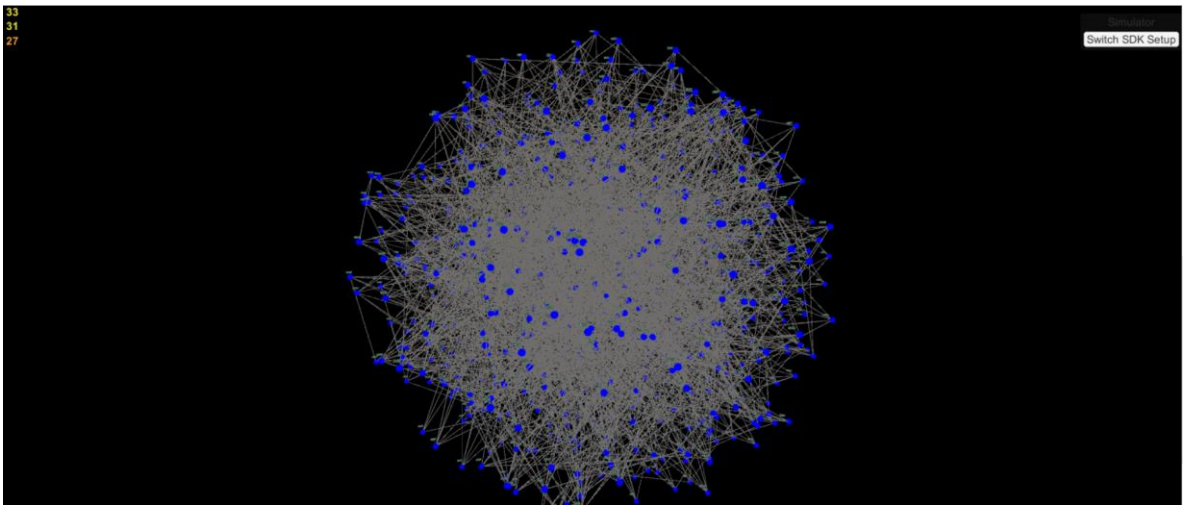

Figura 6.9 Vista global de la red desplegada aleatoriamente en una esfera de 50 metros de radio.

La figura 6.10 muestra la selección múltiple de nodos y aristas utilizando el mando derecho. Se despliega en pantalla solo una región del grafo, resultando en un incremento de los cuadros por segundo. El cambio de color de los elementos se logra mediante el uso de la API de visualización en las clases de interacción.

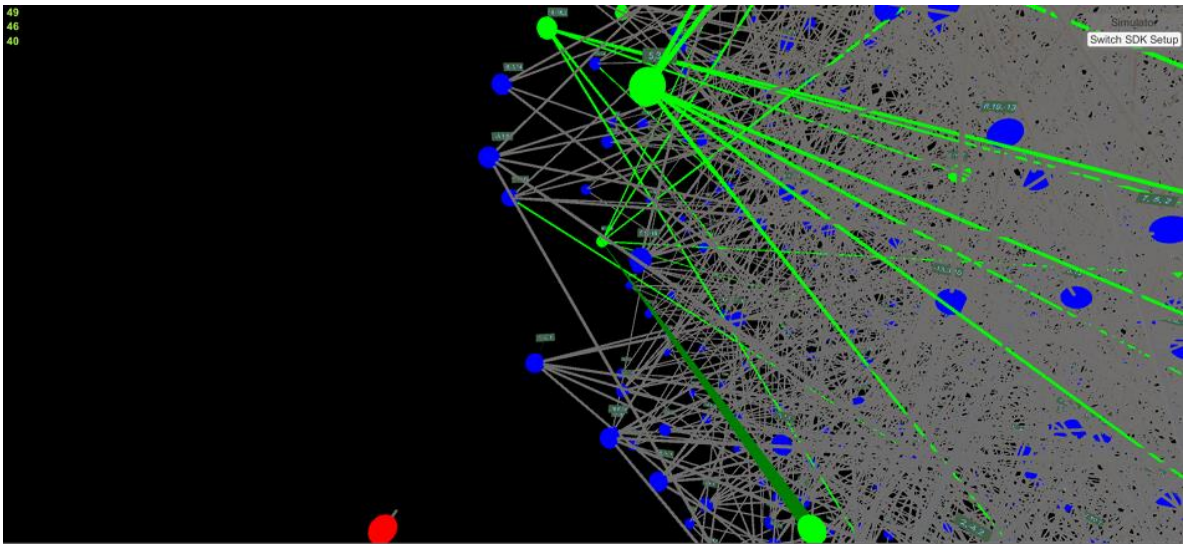

Figura 6.10 Selección múltiple de nodos y aristas mediante rayo utilizando funciones de visualización e interacción.

6.4. Conclusiones

Este capítulo se enfocó a entender la composición de los objetos de Unity y como el renderizado afecta al desempeño del CPU. Se propuso un conjunto de componentes para representar el grafo y modificaciones a los mismos para evitar sobre cargar el CPU. Se implementó una función básica de selección de nodos y aristas, sin embargo, se necesita una función de selección en 3D. La API de interacción requiere de un menú que el usuario pueda usar para interactuar con el grafo y extraer información relevante.

7. PRUEBAS Y RESULTADOS

7.1. Iniciar la ejecución del proyecto.

Al ejecutar el archivo del proyecto el usuario tendrá que seleccionar uno de los dos modos disponibles para la interacción con el grafo como se muestra en la figura 7.1. El primer modo es el simulador con el

cual se puede usar ratón y teclado. El modo simulador dejó de recibir soporte y su funcionalidad es limitada. El segundo modo es el SteamVR en el cual se interactúa mediante el uso de casco y mandos del HTC vive.

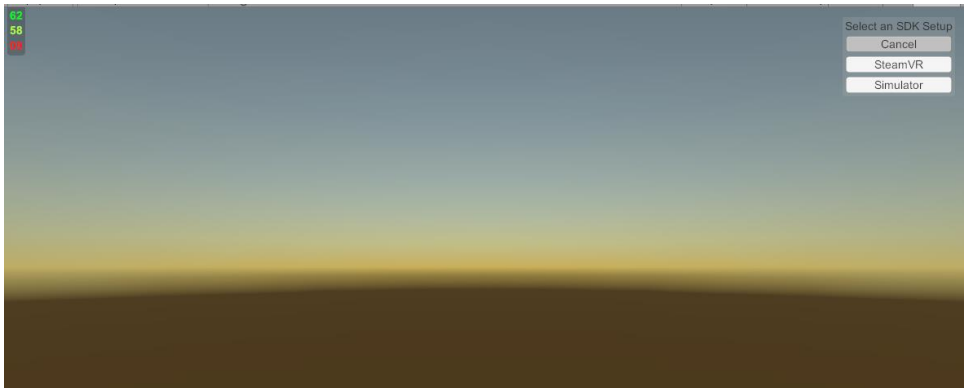

Figura 7.1 Pantalla de inicio

7.2. Cargar un grafo.

Al iniciar el ambiente virtual se tendrá que abrir el panel de interacción, pulsar el botón de nuevo grafo y cargar un grafo mediante el ingreso de un nuevo bolt (herramienta de manejo de contenidos de Amazon *web services*) como se muestra en la figura 7.2**.** El bolt tiene que ser generado por medio de Amazon AWS (Amazon *web services*) el cual apuntara a un grafo creado mediante la base de datos NEO4j, el teclado en pantalla cuenta con botones especiales para facilitar el ingreso del bolt.

Figura 7.2 Ingresar bolt para cargar el grafo

Al seleccionar *enter* el nuevo grafo será cargado y renderizado en el ambiente virtual, listo para iniciar la interacción.

7.3. Control Izquierdo.

En la figura 7.3 se muestra la función principal del control izquierdo es la selección e interacción mediante laser. El láser del control izquierdo es usado para interactuar con el teclado en pantalla, seleccionar aristas y nodos, interactuar con *scrolls* y bonotes de cualquier panel.

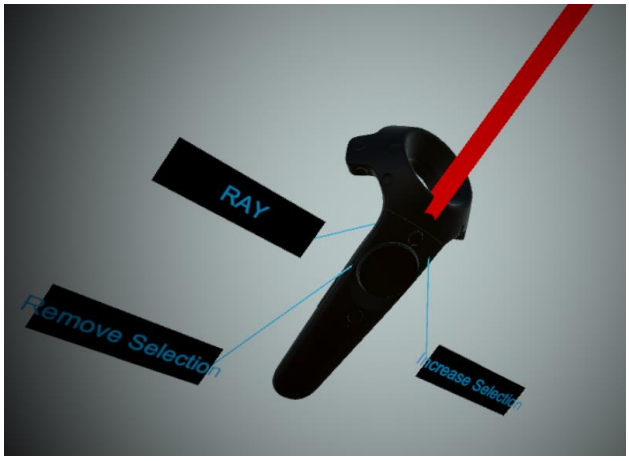

Figura 7.3 Control izquierdo

7.4. Control Derecho.

Actualmente el control derecho tiene dos funcionalidades básicas. La primera funcionalidad se consigue al presionar el gatillo se activará el desplazamiento como se muestra en la figura 7.4. El desplazamiento se realiza utilizando los dos mandos al estilo Superman, mientras más se alejen los controles del casco la velocidad incrementara alcanzando su máxima velocidad con los brazos totalmente estirados. Al presionar nuevamente el gatillo el desplazamiento se detendrá.

Figura 7.4 Control derecho

La segunda funcionalidad es para acceder al panel de interacción. Es necesario pulsar el botón superior del menú circular del control derecho para abrir el panel de interacción. El menú circular tiene otros 6 botones disponibles para abrir otro tipo de paneles como el de visualización.

7.5. Panel de Interacción.

El panel de interacción (figura 7.5) cuenta con múltiples funcionalidades listadas a continuación:

- Configuración de la velocidad de la selección múltiple. La selección múltiple se realiza mediante una esfera casi transparente (figura 7.8) la cual el usuario incrementa su diámetro y todo lo que quede dentro de ella se guardara en una lista de selección tanto para nodos como aristas.
- Configuración de la velocidad de desplazamiento dentro del mundo VR.

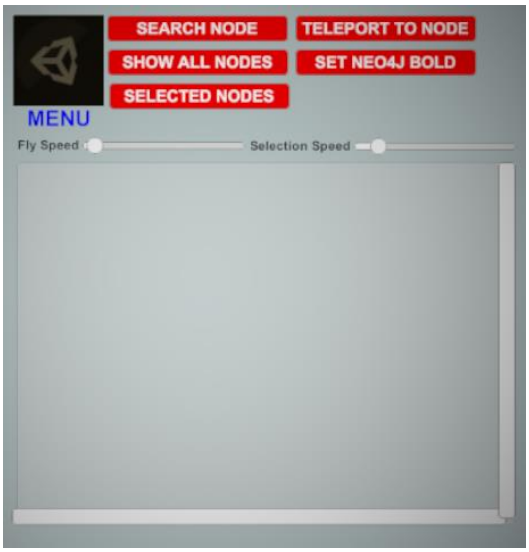

Figura 7.5 Panel de interacción

- Botón de búsqueda de nodo por nombre. En la figura 7.6 se muestra la búsqueda de nodo por nombre activara el teclado en pantalla, con el láser de selección del control izquierdo se pueden teclear las letras, al teclear *enter*, el nodo de existir será seleccionado.

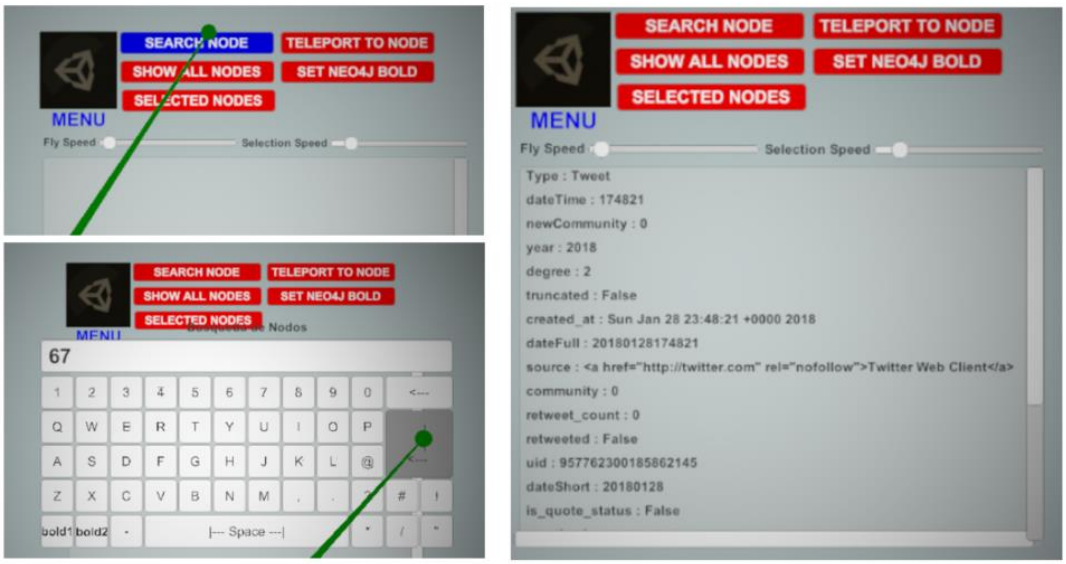

Figura 7.6 Búsqueda de nodo por nombre

- Botón de teletransportación al nodo actualmente seleccionado. Primero se tiene que seleccionar un nodo con el control izquierdo, el nodo cambiara a color verde cuando sea seleccionado. También se puede seleccionar el nodo buscándolo por nombre/id en el panel de búsqueda. Una vez seleccionado al pulsar el botón de teletransportar apareceremos frente al nodo.

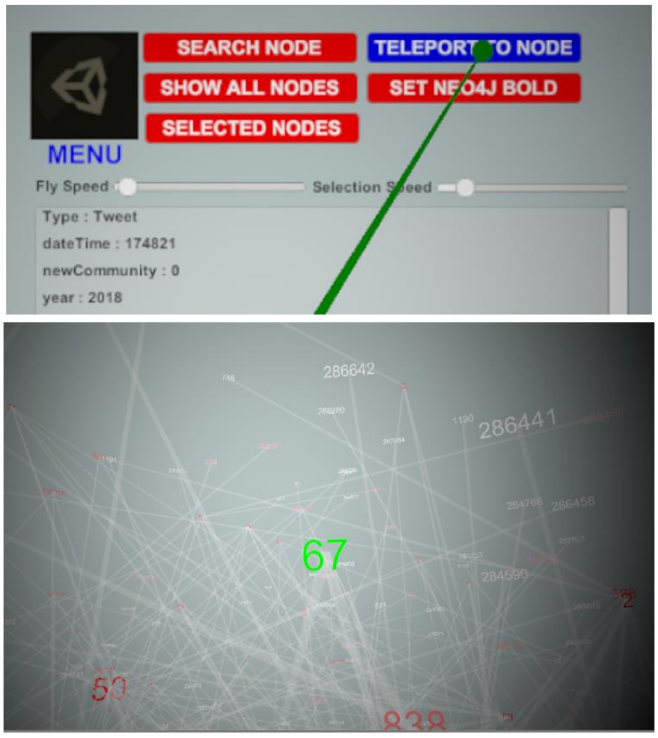

Figura 7.7 Teletransportación a nodo seleccionado

- Botón de mostrar nodos seleccionados y toda su información disponible. La información se mostrará en una lista de desplazamiento con toda la información disponible por cada nodo entregada por la base de datos como se muestra en la figura 7.8.

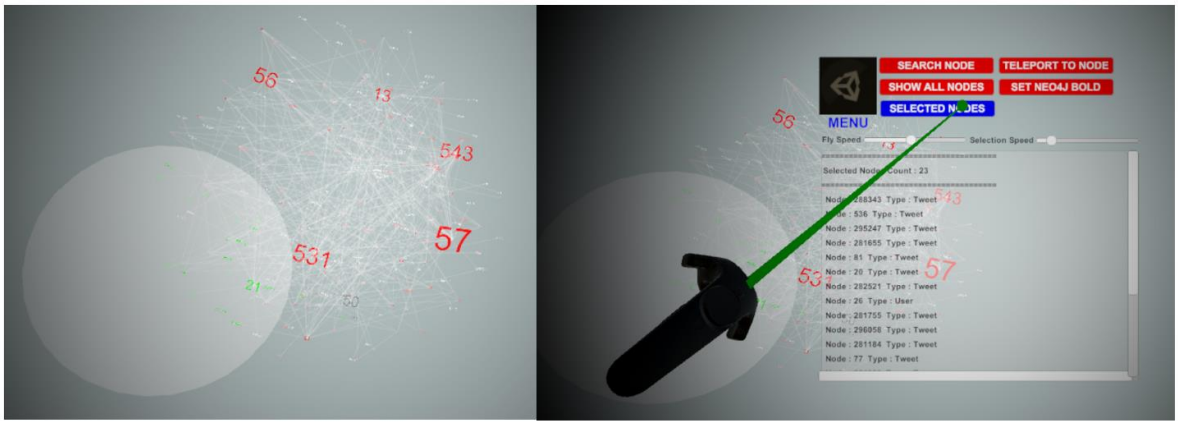

Figura 7.8 Selección múltiple de nodos

- Botón para reiniciar la selección múltiple (figura 7.9). Su función es limpiar las listas de selección controladas por el componente grafo.

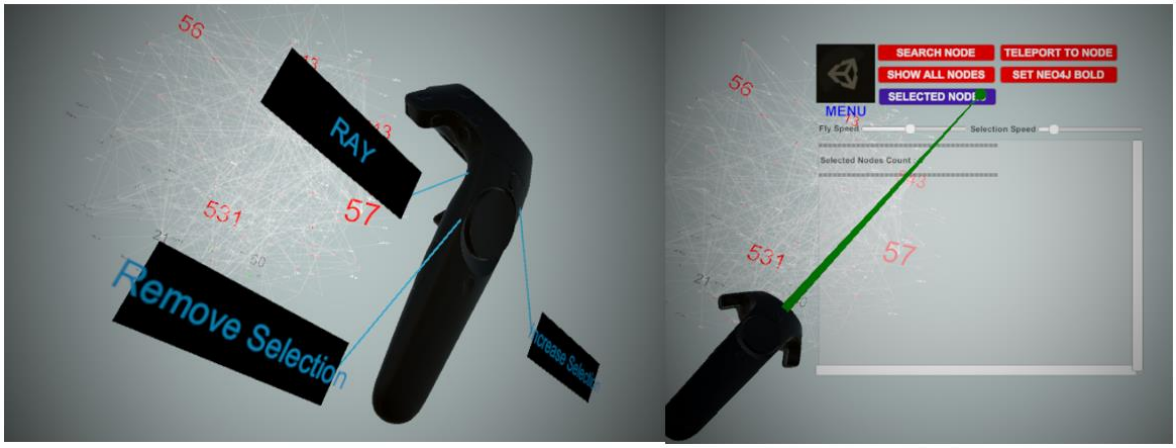

Figura 7.9 Reiniciar selección múltiple

8. CONCLUSIONES Y TRABAJO FUTURO

Se programó una aplicación funcional para el análisis y visualización de grafos en realidad virtual, la cual al ser dependiente de una base de datos de neo4j rompe con la dependencia de procesar archivos de formatos compatibles con topologías de grafos. La API de interacción cuenta con las funciones básicas que el usuario requiere para interactuar con la red y extraer información relevante.

Como trabajo futuro es necesario agregar la sumarizacion de grafos con el propósito de hacer visualmente más comprensible la red y no tener que renderizar todos los nodos y aristas ya que en grafos masivos resultaría en una sobrecarga al CPU y caída de FPS causando en el usuario mareos o nauseas. Es necesario por lo menos un algoritmo de *layout* en 3D para la correcta distribución de los nodos.

BIBLIOGRAFÍA

[1] Neo4j. [Online]. Available: https://neo4j.com/neo4j-graph-database/?ref=home-banner. [Accessed 6 11 2019].

[2] K. Murphy, "Unity Game Engine Review," Game Sparks Technologies Ltd, 6 7 2017. [Online]. Available: https://www.gamesparks.com/blog/unity-game-engine-review/. [Accessed 11 11 2017].

[3] H. Halpin, "Exploring Semantic Social Networks Using," Scotland, UK, 2017.

[4] M. Bastian and S. Heymann, "Gephi: An Open Source Software for Exploring and Manipulating Networks," in *International AAAI Conference on Weblogs and Social Media*, San Jose, CA, 2009.

[5] V. Liluashvili, S. Kalayci, E. Fluder, M. Wilson, A. Gabow and Z. H. Gümüs, "iCAVE: an open source tool for visualizing biomolecular networks in 3D, stereoscopic 3D and immersive 3D," *GigaScience*, vol. 6, no. 8, pp. 1-13, 2017.

[6] A. Robertson, "HTC Vive review," The Verge, 16 9 2016. [Online]. Available: https://www.theverge.com/2016/4/5/11358618/htc-vive-vr-review. [Accessed 19 11 2017].

[7] W. Usher, P. Klacansky and F. Federer, "A Virtual Reality Visualization Tool for Neuron Tracing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 9, pp. 124-136, 2017.

[8] Oculus VR, «Oculus VR,» Oculus VR, 12 3 2012. [En línea]. Available: https://developer.oculus.com/pc/. [Último acceso: 19 11 2017].

[9] MBF Bioscience, "Neurolucida," MBF Bioscience, 15 10 2012. [Online]. Available: http://mbfbioscience.com/neurolucida. [Accessed 11 11 2017].

[10] Microsoft Corporation, "HoloLens," 21 6 2018. [Online]. Available: https://www.microsoft.com/en-us/hololens/hardware. [Accessed 31 10 2019].

[11] S. Beitzel, "Exploring 3D Cybersecurity Visualization," Vencore Labs, Basking Ridge, NJ, USA, 2018.

[12] The Stonefox, "VRTK - Virtual Reality Toolkit," The Stonefox, 13 4 2016. [Online]. Available: https://vrtoolkit.readme.io/. [Accessed 11 11 2017].

[13] Valve Corporation, "SteamVR Plugin," Valve Corporation, 22 6 2015. [Online]. Available: https://www.assetstore.unity3d.com/en/#!/content/32647. [Accessed 11 11 2017].