

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Comunicación y Artes Contemporáneas

Merlinus

Juan Diego Garzón Aguirre

Interactividad y Multimedia

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Licenciado en Interactividad y Multimedia

Quito, 10 de mayo de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Comunicación y Artes Contemporáneas

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

Merlinus

Juan Diego Garzón Aguirre

Nombre del profesor, Título académico

**Mark Bueno M.I.S, Licenciado en
Interactividad y Multimedia**

Quito, 10 de mayo de 2020

DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Nombres y apellidos: Juan Diego Garzón Aguirre

Código: 00138536

Cédula de identidad: 1721054607

Lugar y fecha: Quito, mayo de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETHeses>.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETHeses>.

RESUMEN

Merlinus es un proyecto enfocado en la creación de un videojuego de realidad virtual multijugador con mecánicas innovadoras e inmersivas basadas en los controles del Oculus Quest, un visor de realidad virtual con seis grados de movimiento. El juego consiste en tener un duelo de magia medieval contra otro jugador a través del Internet, usando Unity, el kit de desarrollo de Oculus y Photon como los pilares para la experiencia. El duelo, a su vez, radica en eliminar las fuentes de energía de tu oponente para alcanzar la victoria, usando hechizos ofensivos y defensivos estratégicamente. El prototipo alcanzado sienta las bases para la viabilidad de proyectos que usen este tipo de tecnologías en el país.

Palabras clave: videojuego, realidad virtual, multijugador, mecánicas, gameplay loop, magia.

ABSTRACT

Merlinus is a project focused on the creation of a virtual reality multiplayer videogame with innovative and immersive mechanics based on the Oculus Quest controllers, a virtual reality headset with six degrees of movement. The game consists of a medieval magic duel against another player through the Internet, using Unity, the Oculus development kit, and Photon as the pillars for the experience. The duel, in turn, is comprised of destroying the sources of energy for the opponent to achieve victory, using offensive and defensive spells strategically. The prototype reached is the basis for the viability of projects that use this type of technologies in the country.

Key words: videogame, virtual reality, multiplayer, mechanics, gameplay loop, magic.

TABLA DE CONTENIDO

Introducción	9
Desarrollo del tema	10
Historia	10
Gameplay	10
Metas.	10
Mecánicas.	11
Elementos del juego.....	12
Hechizos.....	12
Tanques de maná.....	14
Castillo.	14
Jugadores.....	15
Estilo gráfico.....	15
Música y audio.....	17
Interfaces y controles.....	17
Target demográfico.....	18
Distribución.	18
Detalles técnicos.	19
Entorno de desarrollo.	19
Plugins.....	19
Loadouts.....	20
Creación de ambiente.....	21
Rigging.....	21
Networking.	22
Scripting.....	23
Conclusiones	24
Referencias Bibliográficas	25

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Gestos de hechizos.....	11
<i>Figura 2.</i> Hechizos ofensivos.	13
<i>Figura 3.</i> Hechizo defensivo.....	14
<i>Figura 4.</i> Tanque de maná.	14
<i>Figura 5.</i> Escena de duelo en el castillo.	15
<i>Figura 6.</i> Menú principal.....	16
<i>Figura 7.</i> Unity, Oculus y Photon.....	20
<i>Figura 8.</i> Rigging del esqueleto.....	22

INTRODUCCIÓN

El prototipo de videojuego creado en el proyecto tiene como título Merlinus, una alusión a la temática de magia medieval presentada. Merlinus es un juego de realidad virtual multijugador con la temática anteriormente mencionada. El jugador tiene duelos uno contra uno contra otro jugador a través del servicio Photon y el Internet, en los que busca destruir las fuentes de energía del oponente para alcanzar la victoria, a la vez que busca defender las suyas. Para esto, se debe usar una combinación de hechizos ofensivos y defensivos de manera estratégica.

El juego se desarrolló con el motor Unity y el kit de desarrollo de Oculus (SDK) para hacerlo funcional con el Oculus Quest, un visor de realidad virtual con seis grados de movimiento, es decir, detección de movimiento y rotación de cabeza y manos (Kuchera, 2019). Esta libertad se usó para las mecánicas principales del juego, el lanzamiento de hechizos, el cual es hecho a través de la posibilidad de la animación de gestos de las manos con los controles del Quest.

El prototipo es una muestra básica del potencial y viabilidad del uso de tecnologías innovadoras y de uso abierto como Unity y el SDK para impulsar una nueva área creativa en el país casi sin costo alguno y la posibilidad de generar videojuegos no tradicionales focalizados en ciertas áreas, dejando otras muy simples o inexistentes, pero sin disminuir la calidad del producto.

DESARROLLO DEL TEMA

Historia

Merlinus no posee una narrativa tradicional puesto que el enfoque es partidas multijugador de corta duración. Sin embargo, el ambiente y recursos usados permite que el jugador experimente un worldbuilding básico de fantasía medieval oscura con elementos mágicos. Se busca que el jugador se vea inmerso en un mundo en el que participa de un torneo académico de magia para determinar el mejor mago del reino. El jugador es un estudiante en una academia de magia, representada por el castillo, y su objetivo final es demostrar que es el mejor al derrotar a sus contrincantes en duelos no letales. Por esta razón, se usan los cristales de maná como un elemento competitivo en el duelo, ya que no se busca herir al oponente. Simplemente, al destruir los cristales de maná, cristales poderosos sacados de las montañas del reino, se elimina la fuente de magia del aprendiz rival, ya que el cristal es el origen de los poderes mágicos de los hechiceros.

Personajes

De igual manera, Merlinus no contiene personajes de una narrativa tradicional, pero se permite cierto grado de personalización de la representación del jugador dentro del juego a través del sistema de loadouts en el que se escoge el tipo de hechizos a usarse en el duelo, intentando incrementar la inmersión del jugador mediante sus elecciones y que sienta que de verdad representa a un estudiante de la academia de magia. De esta manera, todos los jugadores tienen un estilo al momento de realizar el duelo y esto hace que cada uno se convierta en un personaje diferente dentro del universo ficticio.

Gameplay

Metas.

La meta del jugador en cada duelo es destruir las fuentes de energía del oponente, en este caso, tanques de maná, usando su hechizo ofensivo. Al mismo tiempo, debe proteger los suyos utilizando su hechizo defensivo. El primer jugador en destruir todos los tanques del

oponente gana el duelo. Este es el core gameplay loop que el jugador experimenta, y como tal se busca que sea lo más fluido y atractivo posible para que retenga al jugador y lo obligue a continuar jugando, dada la falta de una narrativa que proporcione ese factor de enganche.

Mecánicas.

El jugador lanza hechizos con botones predeterminados del Quest y se usa el sistema de animación de las manos en realidad virtual para proporcionar un gesto distinto y único a cada hechizo para generar más inmersión. El gesto de puño se usa para los hechizos ofensivos, mientras que el gesto de OK se usa para los hechizos defensivos. Cada hechizo tiene un cooldown de habilidad, es decir, solo se puede lanzar un hechizo cada cierto tiempo. Esto previene el spam de hechizos, lo que controla el flujo del gameplay, y añade un nivel estratégico al duelo puesto que el cooldown del hechizo defensivo es más largo que el ofensivo. Asimismo, el hechizo defensivo requiere de las dos manos para ser lanzado, una para apuntar y otra para realizar el gesto de lanzamiento. De esta manera, se tiene que elegir de manera consciente que tanque se protegerá de un ataque enemigo y esperar a la oportunidad correcta de atacar.

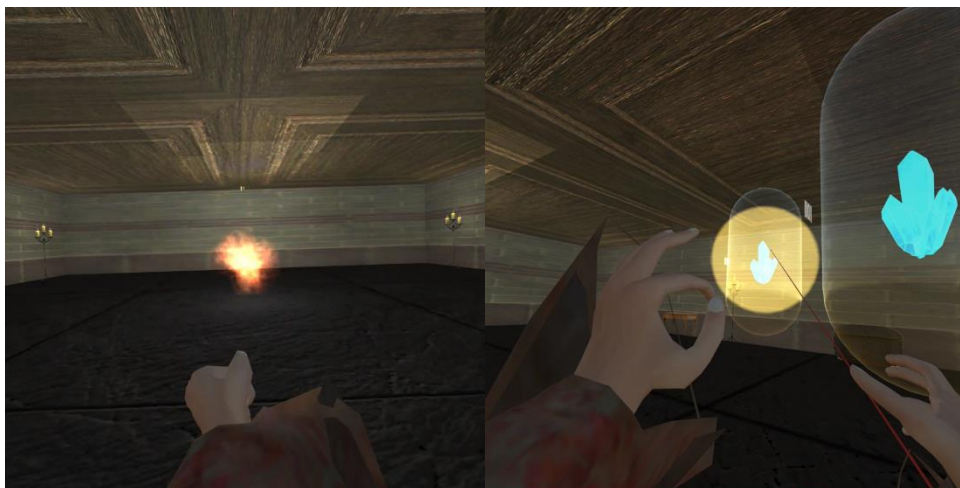


Figura 1. Gestos de hechizos.

Los tanques de maná tienen una cantidad predeterminada de hit points, o puntos de vida. Cada hechizo ofensivo causa una cantidad predeterminada de daño con la que el jugador puede llegar a destruir los tanques del oponente. El sistema de loadout permite

definir que hechizo ofensivo se usa antes de entrar al duelo y genera un nivel estratégico adicional debido a que el jugador debe decidir los hechizos en base a su estilo de juego y características, como cantidad de daño o velocidad del hechizo.

Una vez que uno de los jugadores consigue destruir todos los tanques de su oponente, ese jugador sale victorioso y el duelo se da por finalizado.

Elementos del juego.

Hechizos.

Los hechizos ofensivos en el juego se subdividen en dos categorías. Los hechizos ofensivos se encargan de hacer daño a los tanques de maná del oponente y son lanzados con el gesto del puño. Hay dos tipos de hechizos ofensivos que el jugador puede seleccionar en el sistema de loadouts, cada uno con sus características específicas. La bola de fuego es un hechizo de velocidad alta, pero daño bajo, lo cual lo hace poco predecible pero no tan potente. Por otro lado, la roca es un hechizo de velocidad baja, pero daño alto, lo que lo hace más predecible pero más potente. La elección entre estos dos es parte de la estrategia que el jugador debe usar para ganar el duelo, al igual que la ocasión oportuna para usar el hechizo. Todos los hechizos ofensivos tienen un cooldown predeterminado que impide que se lance un hechizo

repetidamente para que el jugador deba elegir cuando usarlo de manera estratégica y evadir el hechizo defensivo del oponente.

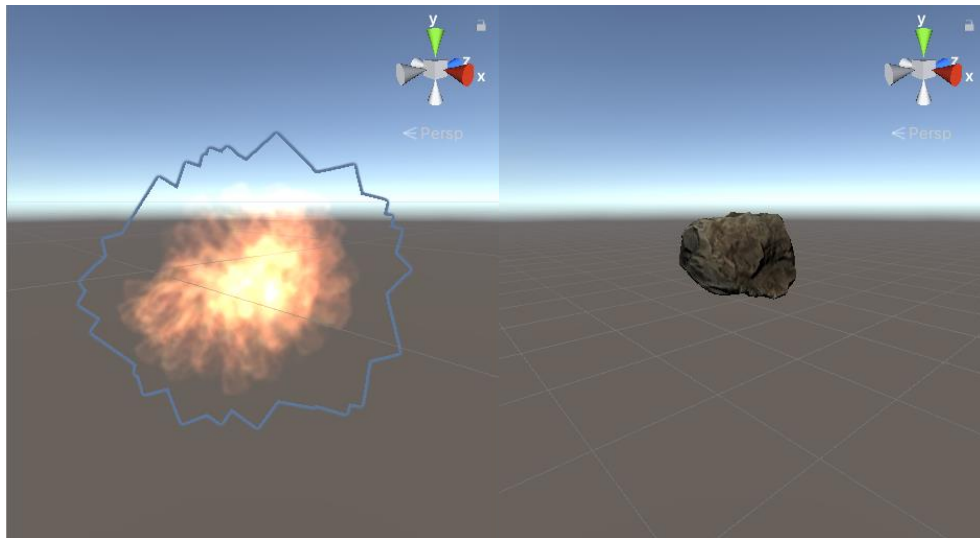


Figura 2. Hechizos ofensivos.

El hechizo defensivo se encarga de proteger los tanques de maná del jugador al proporcionar un escudo que protege de todo daño. El escudo durará un monto de tiempo predeterminado y solo se podrá aplicar a un tanque a la vez. Esto causa que el jugador deba decidir consciente y estratégicamente cuando usar su escudo para negar el daño del oponente sin ponerse en riesgo. De igual manera, todos los hechizos defensivos tienen un cooldown para impedir un lanzamiento indiscriminado, y es más largo que el de los hechizos ofensivos. De esta manera se pone mayor énfasis en defenderse reactivamente y no proactivamente para que el duelo sea fluido y corto en duración, evitando que los jugadores estén a la espera del hechizo defensivo del oponente para atacar.

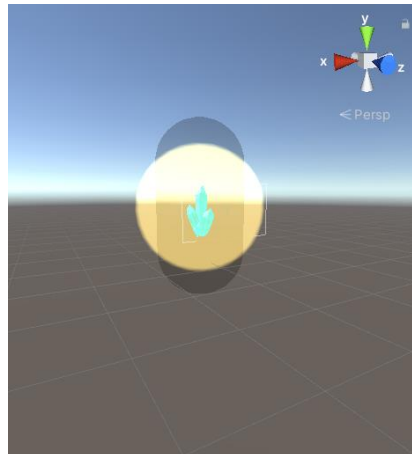


Figura 3. Hechizo defensivo.

Tanques de maná.

Los tanques de maná pertenecientes a cada jugador son cilindros de vidrio que contienen un cristal mágico que proporciona energía a su hechicero respectivo. No tienen un efecto de gameplay fuera de ser el objetivo que se busca destruir para ganar el duelo. Estéticamente, tienen un estilo de fantasía oscura medieval para ayudar con la inmersión de los jugadores obtenido de la tienda de assets de Unity.

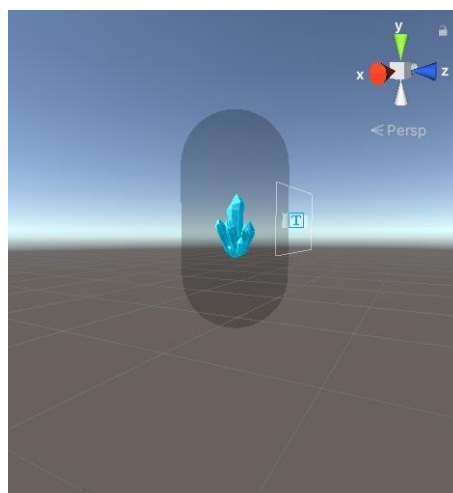


Figura 4. Tanque de maná.

Castillo.

El castillo es el lugar en el que se da lugar a los duelos. Está construido con la misma estética de fantasía oscura medieval para ayudar con la inmersión de los jugadores, pero manteniendo a un mínimo los elementos decorativos para evitar distracciones y mantener espacio para los jugadores y sus respectivos tanques. De

igual manera, los cuartos personales de cada jugador, que funcionan a manera de menú y donde se encuentra el sistema de loadout, también se encuentran en el mismo castillo y, por ende, usan la misma estética. El castillo y los tanques de maná son obtenidos del mismo paquete de assets de la tienda de Unity para mantener consistencia visual.

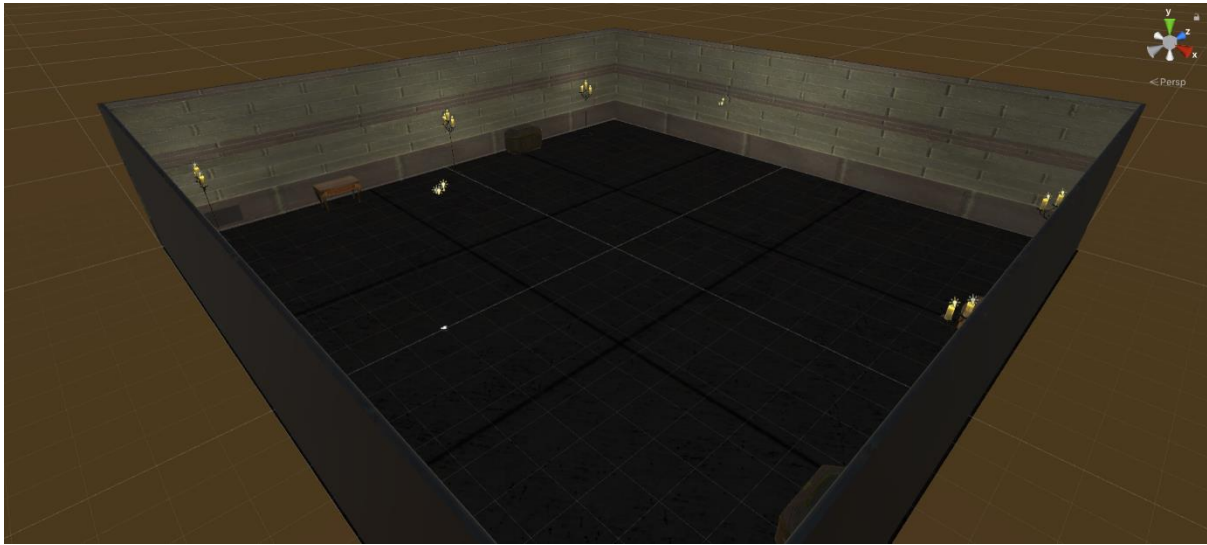


Figura 5. Escena de duelo en el castillo.

Jugadores.

La representación de los jugadores se hace a través de modelos 3D de hechiceros obtenidos de la tienda de assets de Unity. Los modelos tienen esqueletos de animación completamente funcionales, lo que permite usar la animación de manos del SDK de Oculus para generar gestos para los hechizos. La representación de estas animaciones era importante para asegurarse que el hechizo defensivo sea reactivo de manera cómoda al poder ver en tiempo real las acciones de tu oponente, especialmente los gestos que se hace con la mano.

Estilo gráfico.

El estilo gráfico que se usó para los assets de ambiente del juego todos tienen la temática de fantasía oscura medieval con elementos mágicos. Tanto el castillo y todas sus decoraciones como los tanques de maná son realizados con el mismo paquete de assets de la

tienda de Unity para mantener una consistencia visual que facilite la inmersión y la jugabilidad al evitar el sobrecargo de elementos decorativos. La representación de los jugadores sigue la misma línea gráfica, si no el mismo estilo dado que es de un pack de assets diferente al del castillo y los tanques. Todos los assets usados son simples, pero medianamente realísticos. De esta manera, se mantiene la sensación de inmersión de los jugadores, pero se evita que el Quest, un visor con capacidad de hardware limitada, presente problemas de rendimiento al momento de jugar. Finalmente, la tipografía también busca mantener la impresión de ser letras escritas por magos medievales y se usa la misma, Possum Saltare, para todo lo necesario dentro del juego.

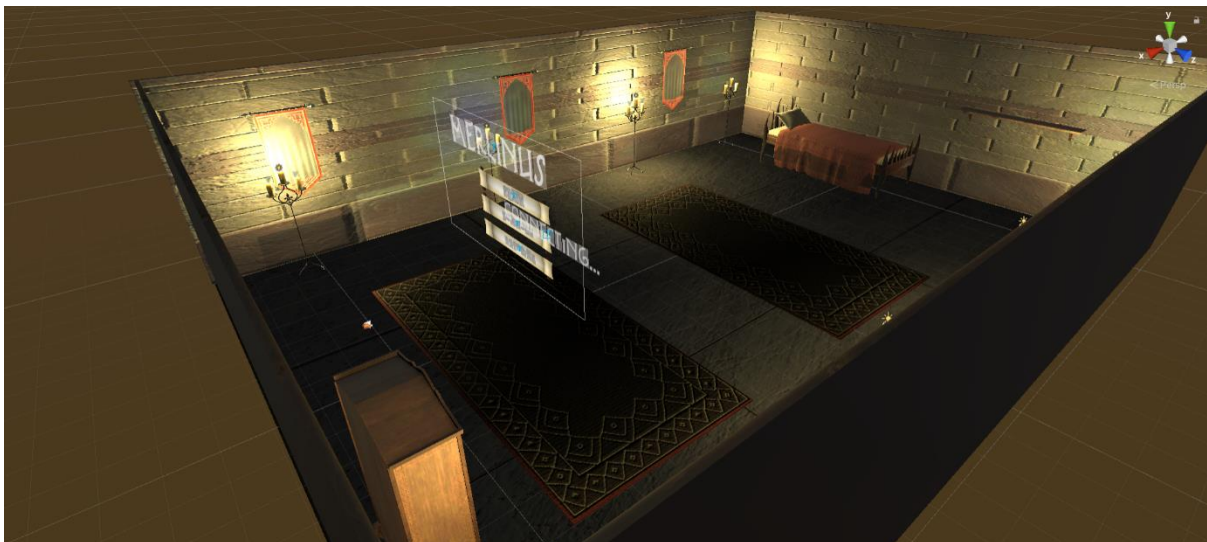


Figura 6. Menú principal.

La iluminación, por su parte, es un elemento difícil de balancear puesto que se necesita encontrar ese equilibrio entre ambiente iluminado solamente por velas y candelabros, pero con suficiente visibilidad para los jugadores. El ambiente es importante para la inmersión y la visibilidad es esencial para que la competitividad del juego no se vea afectada, así como para prevenir los problemas de mareo y náusea muchas veces ocurrentes en la realidad virtual debido a una falta de iluminación adecuada.

Música y audio.

La música del juego funciona como ambiente para los jugadores. Las dos canciones usadas en el juego son de artistas independientes cuyo trabajo es público. La canción del menú es más relajante y calmada para crear el ambiente de que el jugador se encuentra en su cuarto y tiene el control sobre iniciar un duelo y su sistema de loadouts. La canción de duelo, en cambio, es intensa y busca generar un ambiente épico y de batalla para que los jugadores se vean aún más inmersos en la experiencia. Ambas piezas son de carácter medieval con fuertes influencias de folk celta, algo estrechamente relacionado con los mundos de fantasía de la ficción.

Los efectos de sonido del juego buscan, de igual manera, aumentar la inmersión de los jugadores. Efectos para el lanzamiento y contacto de los hechizos crean un mundo auditivo más rico para los jugadores y el efecto del tanque rompiéndose sirve como alerta que puede ser muy útil en el core gameplay loop por su función de dejar saber al jugador que ha perdido un tanque, además de su uso estético.

Interfaces y controles.

Los controles usados por el jugador son los incluidos con el Oculus Quest, que proporcionan detección de movimiento y rotación de las manos, así como cinco botones para cada una. Merlinus hace uso de los botones de puño e índice, también llamado gatillo, para el lanzamiento de hechizos. De igual manera, la posición y rotación de las manos determina el origen del hechizo ofensivo y su dirección, así como el objetivo del defensivo. Para la función de apuntar, tanto como para el hechizo defensivo como para los elementos de interfaz de usuario como las distintas formas de menú existentes, se usa un sistema de point-and-click en el que se rastrea la posición y rotación de las manos para manipular un apuntador láser dentro del juego con el que señalar y seleccionar distintas opciones. Toda la interfaz de usuario es estática o se mueve con la cabeza del jugador para evitar problemas de cinetosis.

El SDK de Oculus se usa para determinar el perfil de la persona en cuanto a altura y el control del esqueleto del modelo 3D de acuerdo a las posiciones y rotaciones de ambas manos. Los controles son lo más simplificados posibles para evitar confusión y frustración, especialmente porque la realidad virtual en sí ya es una tecnología nueva y muchas veces intimidante para un público nuevo.

Target demográfico.

Merlinus está diseñado para ser jugado por jóvenes y adultos de entre 12 y 50 años. Los controles son simplificados para que ambos extremos puedan aprenderlos en un espacio limitado de tiempo, y si bien existen elementos estratégicos más profundos en el juego, también se puede disfrutar de un duelo rápido de manera menos competitiva. Es fácil de aprender, pero difícil de dominar.

La temática igualmente es apropiada para todas las edades y la falta de violencia hace que niños puedan disfrutar del juego sin preocupación parental. La experiencia requiere de mínima actividad física, lo que la hace más accesible a personas de mayor edad y personas con capacidades especiales, así como un requerimiento reducido de espacio, algo importante en los juegos de realidad virtual.

Distribución.

De comercializarse el juego, la distribución se haría a través de la tienda de Oculus dado que es desarrollado para el Quest. Se procedería a cumplir con todos los pasos del proceso discriminatorio de la tienda, incluido una revisión conceptual y técnica del juego, y la creación de todos los elementos necesarios para la publicación, como thumbnails, screenshots y videos promocionales. Después de pasar por el proceso de verificación de Oculus y cumplir todos los requisitos que demanda la tienda, el juego se publicaría por un monto mínimo de cinco dólares al ser un prototipo. Se podría catalogar al juego como en estado de early access,

o acceso temprano, indicando que es un proyecto en proceso y que falta trabajo para completarlo.

Más adelante, con la creación de mejores representaciones para los jugadores, se podría incluir elementos cosméticos a los que se pueda acceder mediante progreso dentro del juego o microtransacciones. Es importante recalcar que cualquier microtransacción dentro del juego sería puramente cosmética y no afectaría de ninguna manera el gameplay de Merlinus. De esta manera, se le ofrece al jugador una manera de personalizar su representación de una manera mucho más profunda.

Detalles técnicos.

Entorno de desarrollo.

El juego fue desarrollado completamente en el motor de juego Unity, específicamente la versión 2019.3.0f6. Varios elementos de Unity fueron usados en la creación del juego, como las interfaces visuales de usuario para los menús y el nuevo paquete de animación y rigging que permitió instanciar avatares de cuerpo completo para los jugadores que sean controlados por los mismos en su totalidad. Además de esto, se escribieron varios scripts de código originales en el lenguaje que usa Unity, C#, para las funciones del core gameplay loop como el manejo de hechizos y la destrucción de tanques.

Plugins.

Dos elementos externos a Unity fueron la base de la creación de Merlinus. En primer lugar, el SDK de Oculus permite desarrollar para el Quest y hacer uso de las funciones de detección de movimiento en seis grados del visor de manera simple. Objetos predeterminados como la función de la cámara en realidad virtual y el rastreo de movimiento y rotación de cabeza y manos fueron sacados del SDK para facilitar el proceso del desarrollo.

En segundo lugar, se usó el motor de networking Photon y su integración a Unity para llevar a cabo toda la funcionalidad de multijugador. Gracias al servicio de Photon se maneja la conexión entre dos jugadores y la sincronización de sus acciones a través del Internet. Se usa scripts predeterminados de networking de Photon, pero altamente modificados para llevar a cabo las acciones necesarias que el juego necesita en su funcionamiento. Desde la conexión inicial, pasando por la instanciación en línea de cuerpos completos para los jugadores y el rastreo en la red de todos sus movimientos, así como de sus hechizos.

La instanciación de jugadores resultó especialmente problemática dada la naturaleza de realidad virtual del juego. Photon no tiene una integración oficial con realidad virtual, por lo tanto, no tiene funciones específicas dirigidas al manejo de representaciones de jugadores con movimiento real como un usuario de realidad virtual. Para solucionar este problema, se debió separar el control del jugador de su cuerpo y animaciones, proceso que se explica a detalle posteriormente.



Figura 7. Unity, Oculus y Photon.

Loadouts.

El sistema de loadouts se logró con scripting original que permite almacenar la elección del jugador en variables estáticas para poder acceder a ellas aún después de la carga de otro nivel. Dependiendo de esta variable, se determina la naturaleza del hechizo ofensivo que se lanza, incluyendo sus características de daño y velocidad. De

igual manera, así se decide que modelo 3D se instancia en línea y localmente a través de Photon, es decir, se establece si una piedra o una bola de fuego debe aparecer cuando el jugador aplasta el botón asociado a un hechizo ofensivo.

Creación de ambiente.

La creación de ambiente se hizo con paquetes de assets gratis de la tienda de assets de Unity. La creación y estructura del castillo y de los tanques de maná fue original usando los modelos de dichos paquetes. El modelo de los jugadores no fue modificado, excepto para hacer un rigging del esqueleto, como se explicó anteriormente. La fuente y la música, usadas en el texto y el audio del juego respectivamente, fueron editadas para su uso en Unity a través de la herramienta de TextMeshPro y AudioSource del motor.

Rigging.

El control del modelo 3D por parte del jugador se genera localmente usando las funciones del SDK de Oculus y el paquete de animación de Unity mencionado anteriormente. Haciendo uso del esqueleto completo del modelo, se usa cinemática inversa para generar movimiento en los brazos y cabeza del modelo 3D de acuerdo al movimiento de las manos del mismo. La cinemática inversa consiste en usar el script predeterminado del paquete para establecer un objetivo y un hint para cada hueso de cada brazo. El objetivo es la mano respectiva y el hint es la posición máxima de flexión del codo. De esta manera, los huesos del antebrazo y el hombro se mueven de acuerdo a la mano seleccionada y se consigue una animación fluida y razonablemente natural. El movimiento de las manos, a su vez, es igualado al movimiento de las manos que se rastrea a través del Quest y los scripts predeterminados del Oculus SDK. Así, el jugador efectivamente controla los brazos y la cabeza del modelo 3D con su movimiento en la vida real.

Finalmente, para conseguir la animación de los dedos de las manos, se usa un script predeterminado que permite guardar posiciones de los dedos como una animación de Unity. Se generan posiciones para default, puño y el gesto de OK, y se usa un árbol de combinación de animaciones de Unity para que la transición entre todas sea lo más fluida posible, aplicándolo a cada mano y especificando que los cambios deben darse cuando el jugador aplaste uno de los botones de hechizos. Esto es básicamente recrear desde cero el sistema que usa el SDK de Oculus para sus manos predeterminadas, con el objetivo de usarlo en las manos del modelo 3D. Luego, todo esto pasa a ser incluido en el networking.

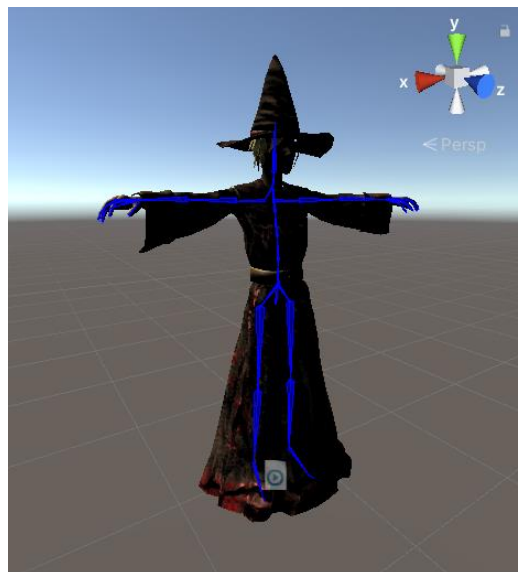


Figura 8. Rigging del esqueleto.

Networking.

El networking de Merlinus tiene muchos pasos. El primer paso es la conexión de jugador a jugador, en la que un jugador busca un room o cuarto en línea al que unirse o, si no existe ninguno, crea uno propio. Al momento de conectarse, se usa la instanciación de Photon para generar las representaciones tridimensionales de los jugadores en la escena, así como sus cuatro tanques respectivos. En el backend, scripts predeterminados de Photon se encargan de manejar la sincronización de los

jugadores al room, al igual que sus movimientos y rotaciones el uno con el otro. El networking de controles y gameplay se hizo con scripting original.

Scripting.

Scripting original se usó para la mayoría de la funcionalidad del juego. Desde modificaciones a scripts predeterminados del SDK de Oculus para asegurarse que el rigging de los esqueletos funcionaba apropiadamente, incluyendo modificaciones muy importantes a los scripts de Photon para controlar la instanciación de jugadores y objetos, así como impedir que los jugadores controlen todas las instancias generadas, es decir, que cada jugador controle su representación y nada más en la escena.

El manejo de input de los jugadores es scripting original que primero chequea que la instancia del jugador sea suya, para luego determinar el lanzamiento de hechizos con los botones de los controles y el movimiento del modelo 3D de acuerdo al movimiento del jugador con el visor de realidad virtual. De igual manera, el manejo de los puntos de vida de los tanques y la colisión de los hechizos con los mismos está hecho a través de modificaciones considerables al script de proyectil que se incluía en el paquete de la bola de fuego. El sistema de loadouts es, por su parte, una modificación original al script de conexión de Photon, el mismo que ya había sido modificado anteriormente para la funcionalidad de instanciación de jugadores y objetos, y el del proyectil de bola de fuego.

En general, el desarrollo del prototipo supuso una cantidad muy fuerte de investigación, dada la falta de experiencia con componente multijugador, y, sobre todo, de prueba y error, debido a la falta de documentación apropiada acerca del tema de multijugador en realidad virtual.

CONCLUSIONES

El inicio del desarrollo de Merlinus fue especialmente complicado debido a la inexperiencia con multijugador. La mezcla de multijugador y realidad virtual resultó ser muy difícil dada la falta de documentación acerca del tema, y, por lo tanto, se requirió grandes cantidades de investigación en este aspecto. Si bien terminó siendo un proceso de aprendizaje importante como un desarrollador de videojuegos, la duración que causó el tener que basarse en un método de prueba y error llevó a que se dedique menos tiempo a otros aspectos importantes del juego, como visuales y variación de gameplay.

Por otro lado, la gran cantidad de diseño y programación requerida para obtener un prototipo con funcionalidades y sistemas completos sirvieron como un ejercicio óptimo de las habilidades obtenidas durante la carrera. El producto final es una muestra del potencial que este tipo de proyectos tienen en el país, más aún cuando se usa tecnologías innovadoras y de carácter tan inmersivo como la realidad virtual. Representa una puerta abierta a una nueva área de creatividad para los diseñadores y desarrolladores interactivos.

Sin embargo, Merlinus está lejos de ser perfecto. El sistema de loadouts podría ser mucho más profundo y la fidelidad visual mucho más aguda. Si bien el gameplay es fluido, para verdaderamente alcanzar ese nivel de enganche de un juego tradicional, se debería implementar un sistema competitivo mediante el cuál los jugadores consigan ese mismo sentido de progreso que está presente en una narrativa tradicional. Todos estos elementos no alcanzaron su potencial máximo debido a la dedicación que demandó el aspecto del multijugador, como se mencionó anteriormente. Esta parte del desarrollo de videojuegos se está volviendo extremadamente importante y debería reforzarse en el currículo de la carrera.

REFERENCIAS BIBLIOGRÁFICAS

- Kuchera, B. (30 de Abril de 2019). *The best part of Oculus Quest: It makes the hard stuff look easy*. Obtenido de Polygon: <https://www.polygon.com/2019/4/30/18518019/oculus-quest-facebook-jason-rubin>
- Oculus. (2019). *Get Started with Oculus in Unity*. Obtenido de Oculus: <https://developer.oculus.com/documentation/unity/unity-gs-overview/>
- Photon Engine. (2016). *Photon Documentation*. Obtenido de Photon Engine: <https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro>
- Unity Engine. (2019). *Unity User Manual*. Obtenido de Unity Engine: <https://docs.unity3d.com/Manual/index.html>