FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

VIDEOJUEGO SIMULADOR BASADO EN EL RECONOCIMIENTO DE FRECUENCIAS MUSICALES EN TIEMPO REAL UTILIZANDO REALIDAD VIRTUAL

AUTOR

EDUARDO ALBERTO LOZA MORENO

AÑO

2017

FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

VIDEOJUEGO SIMULADOR BASADO EN EL RECONOCIMIENTO DE FRECUENCIAS MUSICALES EN TIEMPO REAL UTILIZANDO REALIDAD VIRTUAL

Trabajo de Titulación presentado en conformidad con los requisitos establecidos para optar por el título de Ingeniero en Sistemas de Computación e Informática.

Profesor guía
MSc. Carlos Andrés Muñoz Cueva

Autor
Eduardo Alberto Loza Moreno

Año
2017

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

_____

Carlos Andrés Muñoz Cueva
Magister en Gerencia de
Sistemas C.I: 1712981511

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

_____

Santiago Ramiro Villareal Narváez
Master en Ciencias, Tecnología y Salud mención Informática C.I: 1713980074

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

_____

Eduardo Alberto Loza Moreno C.I: 1723357016

AGRADECIMIENTO

A mi familia, por su amor y su apoyo constante para culminar mi proyecto, a mi novia por toda la confianza puesta en mí en todo momento en el que lo necesite. De manera especial y sincera expreso mi agradecimiento al MSc. Carlos Muñoz mi director de tesis por su valiosa orientación y apoyo, por el tiempo invertido en la dirección y corrección de este proyecto para la culminación del mismo.

Eduardo Loza

DEDICATORIA

Este trabajo de graduación va dedicado para todos los jóvenes innovadores cuyo objetivo es hacer de la tecnología más importante y atractiva a la sociedad.

RESUMEN

Proyecto sobre un videojuego simulador de realidad virtual que, mediante el reconocimiento de las frecuencias de una canción, modifica los aspectos físicos de los objetos que se encuentran alrededor del jugador mientras se encuentra en la simulación, permitiéndole interactuar con las manos el entono virtual.

Esta simulación está realizada en Unity 5.5 y dispositivos como el Oculus Rift DK2 y el Leap Motion, utilizando como lenguaje de programación C# en Visual Studio 2015, la simulación tiene la capacidad de ser totalmente inmersiva y dinámica dependiendo de las frecuencias reconocidas en una canción, estas frecuencias son divididas en grupos para diferenciar las distintas partes de una canción como el bajo, los agudos, etc, estos grupos de frecuencias se encuentran en unidades de Hercios (Hz), por lo que en el script se les toma como unidades numéricas enteras, por otra parte, las animaciones de los modelos dinámicos se controlan por Frames per second (fps) o por unidades de escala, entonces utilizamos los valores numéricos del script para modificar los valores de Frames per second (fps) y unidades de escala en la función de update(), que significa que el cálculo de frecuencias se repetirá 60 veces por segundo convirtiéndose en un reconocimiento a tiempo real y haciendo de la simulación muy agradable a la vista. Luego, utilizando la propiedad trigger de los modelos para reconocer cuando el usuario toca con su mano un objeto, y se genera un modelo 3D que se acopla al módulo de huesos que reconoce Leap Motion al detectar una mano, y por último en el código se utiliza la función OnTriggerEnter() para poder manipular el comportamiento en caso de que el evento de topar un objeto suceda, como resultado se obtiene una simulación con un entorno dinámico que cambia dependiendo de la música, totalmente dinámico e interactivo con las manos, y muy atractivo al usuario.

ABSTRACT

The project is about a virtual reality simulation that works using a song frequency recognition script that allows to change physics aspects of different objects that can be found around the player while he is in the simulator, and can interact with their own hand with the virtual environment.

This simulation is working in Unity 5.5 and different systems like Oculus Rift DK2 and Leap Motion, using C# as a programming language in Visual Studio, the simulation have the capacity of being totally immersive and dynamic depending on the song frequencies, this frequencies are divided in different groups that represent different parts of a song for example the bass, the beat, etc, this frequency groups are in Hercios (Hz), that's why in the script this ones were transformed into numeric int, by the other hand, the animations of the dynamic models works with Frames per second (fps) or with scale units, so using the numeric values of the script we can modify the Frames per second (fps) and scale units values with the update() function, that means the calculation of frequencies will be repeating 60 times per second converting this in a real time recognition and making of this simulation really pleasant for the user. Then, using the trigger property of the models to recognize when the user touch and object with their hand, is generated a 3D model that join the bones module that provide the Leap Motion at the moment it detect a hand, and finally the code use the OnTriggerEnter() function to manipulate the behavior in case the event of touching an object activates, as a result is obtained a simulation with a dynamic environment that changes depending the music, totally immersive and interactive with the hands, and very attractive to the user.

ÍNDICE

1. Introducción

En la actualidad el avance tecnológico ha llegado a un nivel muy alto, y aun en incremento a una velocidad muy rápida, y las empresas buscan utilizar las nuevas tecnologías para generar un mayor valor, la innovación siempre ha llamado la atención y por lo tanto mayores beneficios.

Entre estos avances se encuentra la realidad virtual, el tema actual que logró tener una demanda increíble en los últimos años, y el mercado de videojuegos toma esta tecnología a su favor, haciendo posible una nueva manera de entretenimiento para todas las edades.

Oculus Rift el prototipo más estable de realidad virtual, funciona utilizando el principio del ojo del pez para tener un amplio rango de visión y virtualiza la distancia de los ojos para realizar el efecto de profundidad que en conjunto con Leap Motion, hacen de la realidad virtual una experiencia cada vez más real.

Newzoo es una consultora líder dedicada al mercado de los videojuegos, un mercado muy amplio en los últimos años y ha realizado un estudio a nivel mundial para determinar cuál es el escenario actual de este sector, destacando entre todos los datos obtenidos, el cálculo de que actualmente existen más de 1,200 millones de personas que juegas videojuegos en el mundo ya sea en su celular o en una consola. (Newzoo, 2016)

La industria de los videojuegos es una de las más cambiantes y evolutivas de los últimos años, cada vez es mayor el número de personas que frecuentan el uso de videojuegos como un medio de entretenimiento, pero otras maneras de entretenimiento son las películas, la música y los viajes.

Así mismo grandes empresas como Nintendo están optando por el mercado de los videojuegos móviles, y de realidad virtual, en una entrevista realizada en nintendo NX aseguran la investigación en este campo ya que los videojuegos en realidad virtual serán la innovación más potente y con

más potencial de estos años.

Por otro lado, la música también es una de las industrias más grandes del mundo, existen muchos géneros de música, pero no hay persona en el mundo que no haya escuchado música desde la más primitiva hasta la más compleja, la música es el método más infalible de relajación, estimula el cerebro y propaga la activación cerebral para mejorar el aprendizaje de nueva información, la música es el mejor incentivo en momentos de estrés. (Felix, 1989).

Sin embargo, se han registrado muy pocos casos de poder combinar la música, los viajes y los videojuegos, para lograr un método más innovador de relajación enfocado a una sociedad más moderna.

Esta tecnología se puede aplicar a muchos mercados, desde los videojuegos hasta el marketing en música, este tipo de software es algo muy innovador que no se ha utilizado en el país, gracias a esta tecnología, el interés de la sociedad por la tecnología crecerá aún más, y así poder generar más valor.

Combinar los videojuegos con la música engloba dos de los mercados más grandes a nivel mundial, y la accesibilidad del simulador permite que cualquier persona de cualquier edad pueda jugar sin importar que gusto tenga por la música, y así relacionar con el cliente una manera inmersiva de jugar con la música.

El objetivo General es desarrollar un prototipo de un Simulador de frecuencias musicales a tiempo real, utilizando un dispositivo de realidad virtual (Oculus Rift) y un sensor de manos (Leap Motion), para poder interactuar en la simulación.

Los objetivos específicos son:

- Configurar la integración entre el sensor Leap Motion y Oculus Rift

para poder virtualizar patrones de las manos y observarlas en realidad virtual.

- Implementar el código para reconocer frecuencias y clasificarlo para transformarlo en fotogramas y poder manipular los objetos tridimensionales con estos valores.
- Realizar la simulación inteligente para que reaccione dependiendo de la canción y de la interacción con el entorno.

2. Marco Teórico

2.1 Metodología SUM

Se aplicará la metodología SUM, ya que es la más adecuada para el desarrollo de videojuegos pequeños, nos permite tener diferentes BETA entregables que nos permitirán corregir errores de una mejor manera, se adapta al proyecto en flexibilidad y versatilidad, y nos permite una mejora continua mediante actualizaciones.

La metodología se adapta a la estructura y roles de Scrum, una de las ventajas es la flexibilidad de adaptarse con otras metodologías para obtener una solución a distintas realidades, su alcance en equipos de trabajo es corto, así como el desarrollo de proyecto pequeños, otra ventaja es el alto grado de participación del cliente que permite tener un sistema de acuerdo a las necesidades del mismo.

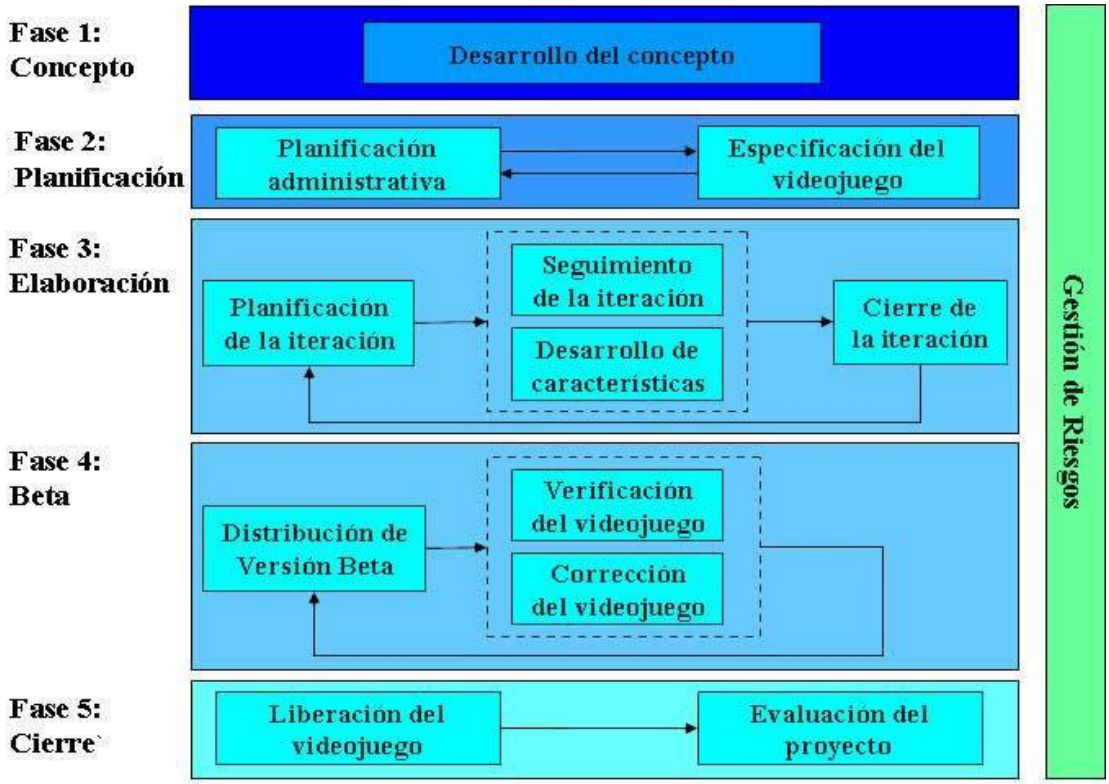

Figura 1. Metodología SUM Desarrollo del Concepto
Adaptado de SUM, 2017.

2.1.1 Concepto

- **Proponer Ideas**

  Se proponen ideas para poder definir una visión y características principales de la simulación que se desea realizar, se utilizan bocetos para representar las ideas.

- **Definir la visión del juego**

  Se define lo que se quiere crear con la simulación, especificando cuales son los objetivos, como se logran estos objetivos, cuáles son los retos y el entorno en el que se desarrolla.

- **Definir genero**

  Se identifica el género basándose en otros títulos del mismo género, este puede ser único o combinado entre distintos tipos de género.

- **Definir gameplay**

  Se identifican el tipo de acciones que se va a realizar en la

simulación.

- **Definir características**

  Se lista las características de la simulación, se las detallan y se define como se las va a implementar.

- **Definir historia y ambientación**

  Se describe un universo de la simulación con más detalle, donde se ubican los personajes principales, motivaciones y como se lograrán objetivos.

- **Realizar pruebas de concepto**

  Se realizan pruebas para mejorar lo mejor posible el concepto de la simulación, es importante tener un concepto bien definido antes de continuar con el proyecto.

Figura 2. Proceso Concepto
Tomado de SUM, 2017.

2.1.2 Planificación

- **Identificar necesidades del proyecto**

  Basándose en el concepto identificar lo necesario para cumplir con el desarrollo del proyecto.

- **Definir equipo de desarrollo**

  Se selecciona a los miembros que van a formar el equipo dependiendo de las habilidades necesarias.

- **Definir contratistas externos**

  Se identifica procesos que no podrán ser realizados localmente, y tendrán que ser tercerizados, identificando el equipo externo que estará en el proyecto.

- **Definir cronograma de elaboración**

  Se define la duración de las etapas de elaboración.

- **Definir cronograma de Beta**

  Se define la duración de las etapas para desarrollar los entregables.

- **Definir cierre del proyecto**

  Se define una fecha estimada de finalización de proyecto.

- **Definir historia y ambientación**

  Se define los hitos de cada fase de desarrollo con su objetivo.

- **Definir objetivos**

  Se definen los objetivos con sus respectivas características que guían al proyecto.

- **Definir criterios de evaluación**

  Se definen los criterios para poder medir si se cumplen los objetivos planteados.

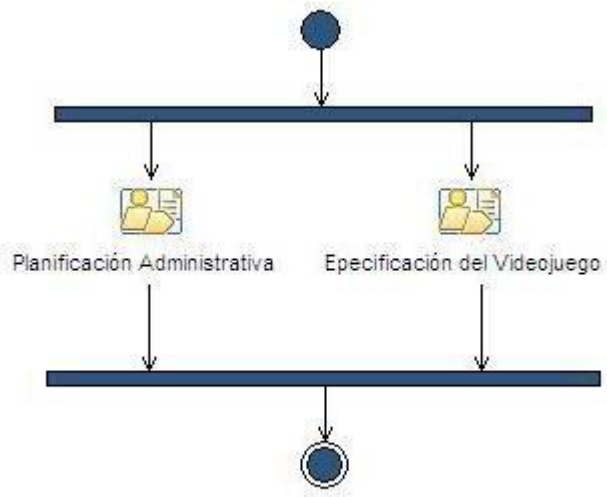

Figura 3. Proceso Planificación
Tomado de SUM, 2017.

2.1.3 Elaboración

- **Seleccionar tarea**

  Se definen y se seleccionan los responsables para las distintas tareas.

- **Ejecutar tarea**

  El responsable seleccionado ejecuta la tarea asignada, esta se considera terminada cuando cumpla con los criterios definidos.

- **Verificar característica**

  Se detecta si las pruebas son satisfactorias para poder validar las tareas realizadas, y se define la característica por completada, en caso de que no sean satisfactorias se crean nuevas tareas para corregir la característica.

- **Manejar problemas**

  Se detectan los problemas e impedimentos para poder progresar en el proyecto, si se detecta un problema se identifica la causa, el impacto y las posibles soluciones.

- **Determinar y comunicar el estado actual del proyecto**

  Se determina el estado del proyecto basándose en la iteración planificada, la comunicación reduce el riesgo de un trabajo en equipo erróneo.

- **Registrar medidas**

  Se registran las métricas que permiten tener visibilidad sobre el estado de la iteración, y se mide la velocidad del equipo para completar las tareas con los objetivos planificados de la iteración.

- **Identificar los elementos positivos y negativos**

  Se extraen los aspectos negativos y positivos que ocurrieron durante la iteración.

- **Evaluar el cumplimiento de los objetivos**

  Se evalúan según los objetivos de la iteración, si no se logró cumplir se debe estudiar la causa, el impacto y cuáles son las acciones a

tomar.

- **Proponer mejoras al proceso**

  Proponer soluciones que puedan resolver los problemas detectados, en caso de que la solución se convierta en una terea se la agregara en la próxima iteración.

- **Determinar cambios**

  Basándose en la iteración se debe determinar qué cambios se realizarán en el plan del proyecto.

- **Realizar cambios**

  Se realizan los cambios que se determinaron y que son necesarios.

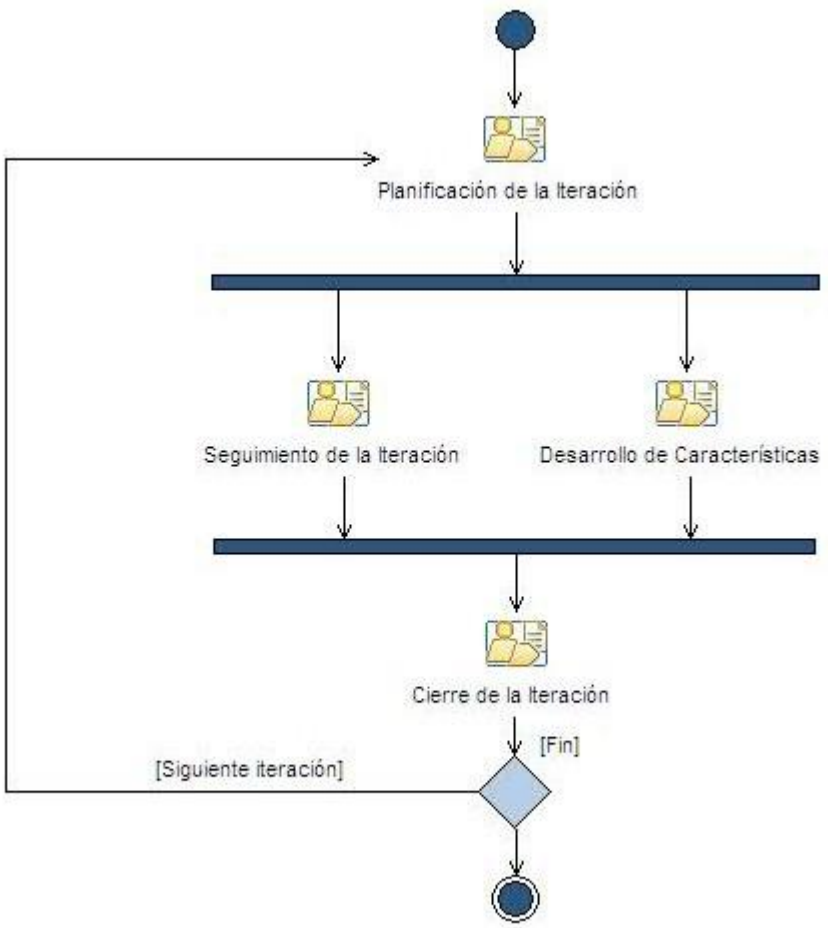

Figura 4. Proceso de Elaboración
Tomado de SUM, 2017.

2.1.4 Beta

- **Definir aspectos funcionales**

  Se definen los aspectos que serán tenidos en cuenta para verificar la simulación.

- **Definir medios de distribución**

  Se define como se va a distribuir la versión beta.

- **Definir como se reportan los errores**

  Se define cuáles van a ser los medios por el cual se van a reportar errores de la beta.

- **Definir verificadores beta**

  Se va a determinar el equipo que estará encargado de encontrar errores y corregirlos.

- **Determinar y comunicar el estado actual del proyecto**

  Se determina el estado del proyecto basándose en la iteración planificada, la comunicación reduce el riesgo de un trabajo en equipo erróneo.

- **Evaluar y verificar**

  Se va a evaluar la simulación y verificar la beta.

- **Reportar resultados**

  Se reportan al equipo de desarrollo los errores y resultados que se encontraron en la evaluación.

- **Evaluar cambios**

  Se realizan cambios basándonos en los resultados de las evaluaciones, y se clasifican.

- **Priorizar cambios**

  Con los cambios clasificados se puede realizar la priorización dependiendo del impacto que pueda causar.

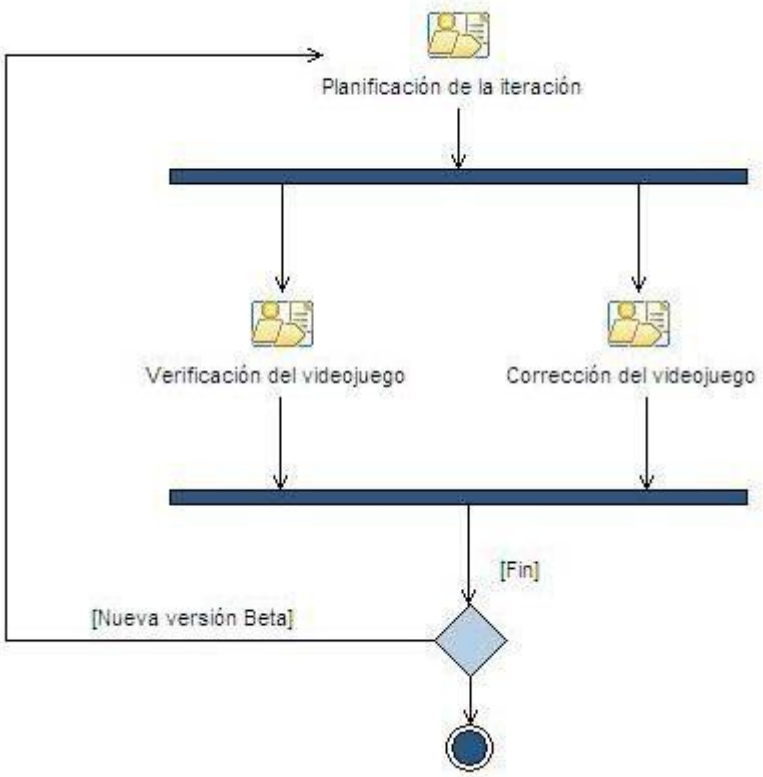

Figura 5. Proceso Beta
Tomado de SUM, 2017.

2.1.5 Cierre

- **Definir entregable**

  Se definen los componentes finales que va a tener el entregable.

- **Realizar entregable**

  Se realizan tareas necesarias para generar un entregable con todos los elementos incorporados.

- **Validar entregable**

  Se entrega al cliente el entregable final para ser evaluado.

- **Registrar lecciones aprendidas**

  Se realizan conclusiones en base a la evaluación y se registran las lecciones que poder ser útiles en futuros proyectos.

- **Proponer mejoras a la metodología**

Se realizan modificaciones a la metodología para que se ajuste más al equipo.

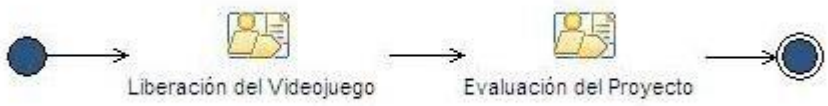

Figura 6. Proceso de Cierre
Tomado de SUM, 2017.

2.1.6 Gestión de riesgos

- **Identificar riesgos**

  Se identificar los posibles riesgos del proyecto.

- **Evaluar riesgos**

  Se define el impacto que tienen los riesgos sobre la parte de desarrollo y determinar el momento que podría dejar de existir en el proyecto.

- **Determinar estrategias para mitigar los riesgos**

  Se determinan las estrategias que se van a utilizar para mitigar estos riesgos.

- **Establecer planes de contingencia**

  Se realizan planes de contingencia para los riesgos más graves del proyecto.

- **Monitorear riesgos**

  Se monitorean y se mitigan los riesgos que se hayan detectado y se aplican los planes de contingencia en caso de ser necesario.

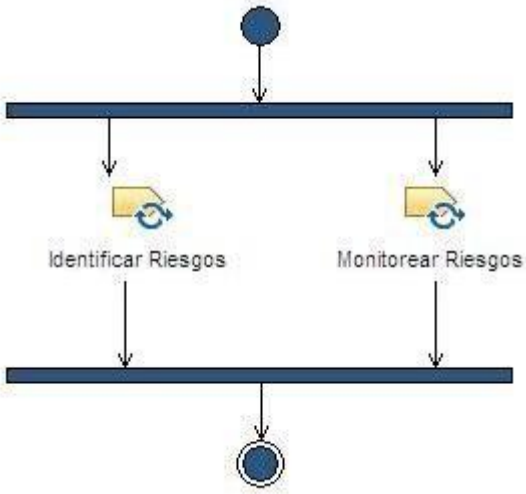

Figura 7. Proceso de Gestión de Riesgos
Tomado de SUM, 2017.

2.2 Herramientas

Para el desarrollo de este proyecto se utilizaron las siguientes herramientas:

2.2.1 Visual Studio 2015

Como entorno de desarrollo integrado, en lenguaje de programación C#, una herramienta que permite la comunicación entre estaciones de trabajo, en este caso para utilizarla con una integración con Unity y poder complementar el desarrollo.

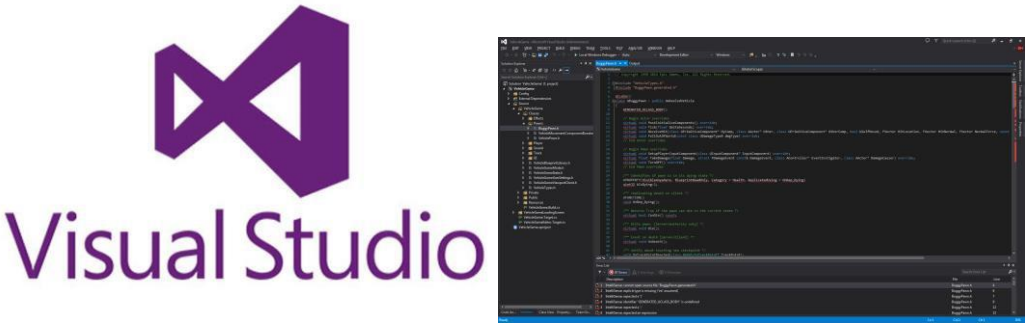

Figura 8. Visual Studio
Adaptada de Microsoft, 2017.

2.2.2 Unity

Como motor gráfico multiplataforma creado por Unity Technologies, su amplia accesibilidad de plataformas lo convierte en el motor gráfico más utilizado de los últimos años, sus instancias se acoplan a Cinema4D y Visual Studio de una manera que cualquier cambio realizando en estos programas, se actualiza automáticamente sin necesidad de re-importar.

Figura 9. Unity 5.5
Adaptada de Unity, 2017.

2.2.3 Cinema4D

Para la creación de animaciones 3D, desarrollado por Maxon, permite modelado, texturación y animación. Su mayor virtud es su alta velocidad de renderizado con una interfaz altamente personalizable y flexible, con una curva de aprendizaje muy buena.

Figura 10. Cinema 4D
Adaptada de Maxon, 2017.

2.2.4 MagicaVoxel

Para el modelado de voxel 8-bit, es una herramienta muy liviana e interactiva para crear y renderizar a una gran velocidad modelos simples de 8-bit y poderlos exportar a Cinema4D para hacer el texturizado.

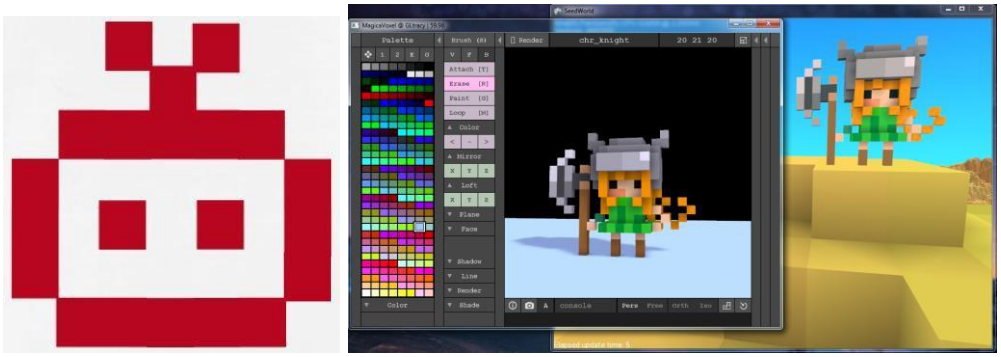

Figura 11. MagicaVoxel
Adaptada de Ephtracy, 2017.

2.2.5 Oculus Rift

El Oculus Rift es un dispositivo de realidad virtual que realiza el trabajo de virtualizar la distancia de los ojos para hacer creer al cerebro humano que lo que observa se encuentra tridimensional, utiliza la base del ojo del pez, de esta manera el lente permite tener un amplio rango de visión, incluye una cámara tracking que permite saber en qué lugar se encuentra el dispositivo y d esta manera utilizar esa información en la simulación.

Figura 12. Oculus Rift DK2
Adaptada de Oculus, 2017.

2.2.6 Leap Motion

El Leap Motion es un dispositivo que funciona mediante sensores infrarrojos que identifican las articulaciones de la mano para poder virtualizar un modelo 3d que actúa con puntos referenciales en cada articulación, se lo puede importar como modelo 3d interactivo para poder utilizarlo en Unity y combinarlo con Oculus Rift.

Figura 13. Leap Motion
Adaptada de Leap Motion, 2017.

2.2.7 Orion

Es un software desarrollado y contribuido por la comunidad de Leap Motion, que permite modificar la recepción de los sensores y así poder utilizarlo en distintas orientaciones para hacer posible la integración con otros dispositivos de realidad virtual.

Figura 14. Orion
Adaptada de Leap Motion, 2017.

2.2.4 Vuforia

El software de uso educacional libre que permite el reconocimiento de patrones utilizando la cámara del dispositivo, y de esta manera poder identificar en la base de datos una indexación con estos patrones para poder ser utilizadas en distintos programas como en este caso Unity.

Figura 15. Vuforia
Adaptada de Vuforia, 2017.

3. Diseño e Implementación

3.1 Concepto

3.1.1 Definición

Tabla 1. *Proceso de Definición*

Idea Propuesta	Se propone el objetivo del simulador, como concepto principal el uso de graficas 8-bit tridimensionales por su capacidad de renderizar rápido, ya que se utilizará Oculus Rift y se necesita la menor latencia posible y la mayor cantidad de frames por segundos del simulador.
Visión del juego	La visión del juego es demostrar de una manera totalmente nueva e interactiva la integración con la música favorita del jugador para tener una nueva manera de entretenimiento.
Género	El género es de tipo simulación ya que el objetivo no es el de un videojuego convencional, sino de demostrar un comportamiento basado en la música.
Gameplay	Para el gameplay se utilizará Leap Motion para poder interactuar con la simulación a tiempo real.
Características	La simulación se caracteriza por reconocer las frecuencias del celular a tiempo real y modificar todo el ambiente dependiendo de cómo reaccionen estas frecuencias.
Historia y ambientación	La simulación no cuenta con una historia y la ambientación se realiza en distintos planos con gráficos voxel y gráficos avanzados dependiendo de la ambientación.
Pruebas de concepto	Se descarta la posibilidad de que la simulación no sea agradable al usuario ya que se utiliza la música de gustos propios de cada jugador y se confirma el tiempo indicado.

3.2 Planificación

3.2.1 Planificación Administrativa

Tabla 2.*Planificación administrativa*

Necesidades del proyecto	Para el desarrollo del proyecto se necesita como objetivo principal tener conocimiento sobre programación en Unity y un equipo de desarrollo que cumpla con características potentes para el desarrollo, como secundario se propone conocimiento en el uso de software de modelado 3D.
Equipo de desarrollo	Por el tamaño del proyecto el equipo de desarrollo constara solamente de una persona, el cual programara, diseñara y controlara la funcionalidad de la simulación en su totalidad.
Contratistas Externos	El proyecto no tiene la necesidad de tercerizar ningún proceso en la elaboración, por lo tanto, no se necesitará de contratistas externos.

3.2.2 Cronograma

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Presentación a la Comisión	1 día	jue 02/02/17	jue 02/02/17	
2		Aprobación del Proyecto	3 días	vie 03/02/17	mar 07/02/17	1
3		Configuración de Equipos	20 días	mié 08/02/17	mar 07/03/17	2
4		Marco Teorico	20 días	mié 08/02/17	mar 07/03/17	
5		Implementación de reconocimiento de periféricos	10 días	mié 08/03/17	mar 21/03/17	4
6		Reconocimiento y clasificación de frecuencias musicales	5 días	mié 22/03/17	mar 28/03/17	5
7		implementacion de algoritmo de reconocimiento en la simulacion	15 días	mié 29/03/17	mar 18/04/17	6
8		Interfaz gráfica y modelado 3d del simulador	30 días	mié 08/03/17	mar 18/04/17	3
9		Pruebas y regulaciones finales de las frecuencias en la simulación	20 días	mié 19/04/17	mar 16/05/17	8

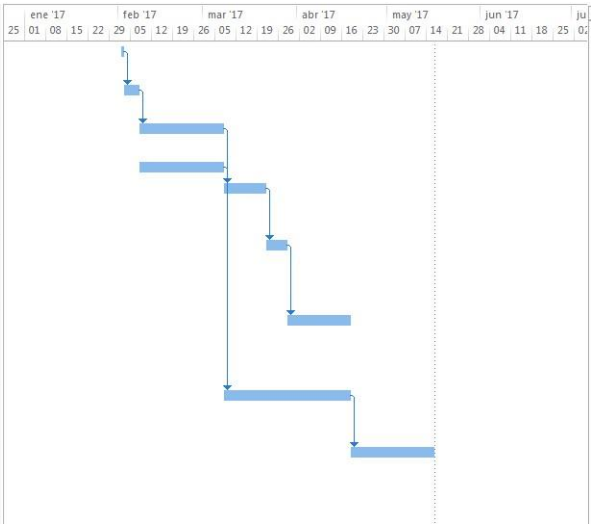

Figura 16. Cronograma

3.2.3 Objetivos

Tabla 3.*Objetivos Generales*

OG01 configuración del ambiente de desarrollo	
Definición:	Se debe configurar la importación entre Unity y el resto de software que se va a utilizar.
Criterio de Evaluación:	Debe funcionar la importación en Unity.
OG02 configuración de periféricos	
Definición:	Se debe configurar el hardware extra que se va a utilizar como son el Oculus Rift, y el Leap Motion.
Criterio de Evaluación:	Debe funcionar en el editor de Unity con prueba de render.
OG03 Diseño del ambiente	
Definición:	Se diseña el ambiente por el cual se va a desenvolver la simulación.
Criterio de Evaluación:	Ambientación completa y funcional.
OG04 Reconocimiento de frecuencias	
Definición:	Se desarrollará el script con funcionalidad que identificará a tiempo real las frecuencias musicales detectadas por la entrada de micrófono (aux).
Criterio de Evaluación:	El script reconoce las frecuencias de la canción a tiempo real.
OG05 Gameplay	
Definición:	Se configura y diseña el gameplay utilizando las configuración y scripts ya realizadas.
Criterio de Evaluación:	Funcionamiento completo del gameplay.
OG06 Desarrollo de interfaces	
Definición:	Se diseña las interfaces que van a ser utilizadas en la simulación.
Criterio de Evaluación:	Interfaces amigables y funncionando.

OG07 Controlador de realidad virtual	
Definición:	Se configura el controlador que permite utilizar la simulación en realidad virtual.
Criterio de Evaluación:	Ambientación funcionando en Oculus Rift.
OG08 Migración con realidad aumentada	
Definición:	Se realiza una migración de la simulación con los componentes necesarios para funcionar en realidad
Criterio de Evaluación:	Ambientación funcionando en aplicativo de celular
OG09 Entregable final	
Definición:	Se realiza un entregable que permita.
Criterio de Evaluación:	Ambientación funcionando en aplicativo de celular

3.2.4 Características

Tabla 4.*Características especificas*

OE01 Configuración Unity		
Roles	Equipo de desarrollo	**Prioridad: 150**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el software de Unity con licencia gratuita para ubicar en entorno 3D, creando un proyecto con estructura para poder realizar la integración con otros programas.	
OE02 Configuración Orion		
Roles	Equipo de desarrollo	**Prioridad: 160**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el software de Orion para poder utilizar una implementación con Leap Motion y poder calibrar el dispositivo.	
OE03 Configuración Cinema4D		
Roles	Equipo de desarrollo	**Prioridad: 200**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el software funcionando en el computador donde se realizarán los diseños de la simulación.	
OE04 Configuración Visual Studio		

Roles	Equipo de desarrollo	**Prioridad: 170**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el Visual Studio para que se integre con Unity, y de esta manera poder utilizarlo directamente como complemento con el código de la simulación desde Unity.	
OE05 Configuración Magica Voxel y Vuforia		
Roles	Equipo de desarrollo	**Prioridad: 180**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se Realiza el registro en el portal de Vuforia con licencia gratuita y se configura Magica Voxel para poder modelar voxels en 3D.	
OE06 Configuración Oculus Rift		
Roles	Equipo de desarrollo	**Prioridad: 210**
Entrada	OG02 configuración de periféricos	
Definición	Se configura el Oculus Rift y se calibra para poder funcionar en el computador.	
OE07 Configuración Leap Motion		
Roles	Equipo de desarrollo	**Prioridad: 220**
Entrada	OG02 configuración de periféricos	
Definición	Se configura el Leap Motion y se calibra con el software Orion previamente instalado para poder funcionar en conjunto con el Oculus Rift.	
OE08 Configuración de funcionamiento de periféricos en Unity		
Roles	Equipo de desarrollo	**Prioridad: 230**
Entrada	OG02 configuración de periféricos	
Definición	Se configura la integración entre Oculus Rift y Leap Motion como un modelo prefab en Unity y poder utilizarlo en la simulación.	
OE09 Diseño de la plataforma de la simulación		
Roles	Equipo de desarrollo	**Prioridad: 250**
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña un modelo de la plataforma por la que se va a interactuar en la simulación desde magicavoxel.	
OE10 Diseño del entorno de la simulación		
Roles	Equipo de desarrollo	**Prioridad: 260**
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña el entorno por partes en mágica voxel, se lo une y se lo optimiza en Cinema4D para poder importarlo en Unity	
OE11 Diseño de personajes		
Roles	Equipo de desarrollo	**Prioridad: 270**
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña los personajes principales y secundarios que se podrán observar en la simulación.	
OE12 Diseño de modelos dinámicos		

Roles	Equipo de desarrollo	**Prioridad: 280**
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña los modelos que van a ser dinámicos, con el complemento de script y superficies en trigger que funcionaran con el script de conocimiento.	
OE13 Animaciones primarias y secundarias		
Roles	Equipo de desarrollo	**Prioridad: 290**
Entrada	OG03 Diseño del ambiente	
Definición	Se realizan las animaciones en los modelos primarios y	
	secundarios que no van a reaccionar con el script.	
OE14 Iluminación de la simulación		
Roles	Equipo de desarrollo	**Prioridad: 295**
Entrada	OG03 Diseño del ambiente	
Definición	Ya que al inicio de la simulación se encuentra en el espacio se desarrolla la contra iluminación, pero en el tercer escenario se encuentra en la ciudad donde necesita	
OE15 Desarrollo de script para reconocer frecuencias a tiempo real		
Roles	Equipo de desarrollo	**Prioridad: 300**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla en lenguaje de programación C# el reconocimiento de frecuencias utilizando la librería spectrum.	
OE16 Desarrollo de script para modificar propiedades de modelos		
Roles	Equipo de desarrollo	**Prioridad: 310**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla en lenguaje de programación C# el comportamiento de los modelos dinámicos utilizando las informaciones obtenidas del script de frecuencias.	
OE17 Desarrollo de script para reconocimiento de música por AUX		
Roles	Equipo de desarrollo	**Prioridad: 320**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla en lenguaje de programación C# el reconocimiento del dispositivo que va a reproducir la canción mediante el cable AUX.	
OE18 Desarrollo complemento Audio Source		
Roles	Equipo de desarrollo	**Prioridad: 330**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla el complemento del audio source con el script de frecuencias, y para reproducir la música utilizando el script de reconocimiento AUX.	
OE19 Administración de colisiones		
Roles	Equipo de desarrollo	**Prioridad: 350**
Entrada	OG05 Gameplay	

Definición	Debido al dispositivo de Leap Motion hay modelos que pueden ser tocados y que interactúan con el entorno, esto significa que algunos modelos necesitaran utilizar propiedades físicas que les permitan realizar estas acciones.	
OE20 Administración de triggers		
Roles	Equipo de desarrollo	**Prioridad: 360**
Entrada	OG05 Gameplay	
Definición	Hay modelos del entorno que no necesitan ningún tipo de interacción o solo necesitan interacción de control y validación, por lo tanto, se les aplica propiedades físicas	
	de triggers.	
OE21 Definir interacciones		
Roles	Equipo de desarrollo	**Prioridad: 370**
Entrada	OG05 Gameplay	
Definición	Se debe realizar scripts que controlen el comportamiento de la interacción del modelo de las manos controlado por Leap Motion, con el entorno que se puede topar o activar.	
OE22 Desarrollo interfaces dispositivo móvil		
Roles	Equipo de desarrollo	**Prioridad: 410**
Entrada	OG06 Desarrollo de interfaces	
Definición	Se desarrollan las interfaces para dispositivos móviles, con las propiedades correctas para que sean responsive a cualquier dispositivo.	
OE23 Administración de controlador general de la simulación		
Roles	Equipo de desarrollo	**Prioridad: 420**
Entrada	OG07 Controlador de realidad virtual	
Definición	Este controlador tiene la funcionalidad de administrar los demás scripts y controlar la jugabiilidad, este se encuentra vinculado al dispositivo de realidad virtual para que se ejecute las interacciones con el usuario.	
OE24 Administración de controlador de integraciones		
Roles	Equipo de desarrollo	**Prioridad: 430**
Entrada	OG07 Controlador de realidad virtual	
Definición	Se desarrolla controlador que permite unir todos los periféricos necesarios como los dispositivos de realidad virtual con el reconocimiento de las manos, y la entrada de AUX, para que no causen conflictos.	
OE25 Administración de controlador de validación		
Roles	Equipo de desarrollo	**Prioridad: 440**
Entrada	OG07 Controlador de realidad virtual	

Definición	Se desarrolla un controlador que permita validar estados de los objetos activados, desactivados, en ejecución, y cancelados, esta validación se aplica con los modelos dinámicos y los estáticos dependiendo a la interacción que tengan con el usuario o con las frecuencias.	
OE26 Exportación de modelos		
Roles	Equipo de desarrollo	**Prioridad: 450**
Entrada	OG08 Migración con realidad aumentada	
Definición	Se deben exportar los modelos dinámicos, estáticos, y de entorno que se puedan utilizar ya que no se tiene tanta capacidad de renderizado, por lo que se exporta solo lo necesario.	
OE27 Creación de Patrón de reconocimiento y base de datos		
Roles	Equipo de desarrollo	**Prioridad: 460**
Entrada	OG08 Migración con realidad aumentada	
Definición	Utilizando Vuforia creamos un patrón que sea amigable	
	referente a la simulación, y realizar una base de datos con los patrones que vamos a utilizar, esta base de datos se la exporta como asset para poderla importar	
OE28 Migración a realidad aumentada con Unity		
Roles	Equipo de desarrollo	**Prioridad: 470**
Entrada	OG08 Migración con realidad aumentada	
Definición	Unimos toda la base de datos, los modelos, y los scripts para que funcionen en un nuevo proyecto, realizamos un cambio de plataforma para celulares móviles, y generamos el Android Application Package (apk) que funcionara con el dispositivo.	
OE29 Pruebas finales en aplicativo de realidad virtual		
Roles	Equipo de desarrollo, usuarios	**Prioridad: 500**
Entrada	OG09 Entregable final	
Definición	Se realizan pruebas para ver que toda la simulación se encuentre en completa funcionalidad.	
OE30 Pruebas finales en aplicativo de realidad aumentada		
Roles	Equipo de desarrollo, usuarios	**Prioridad: 510**
Entrada	OG09 Entregable final	
Definición	Se realizan las últimas pruebas de funcionalidad sobre la aplicación de dispositivos móviles con realidad aumentada.	
OE31 Ejecutable final de escritorio		
Roles	Equipo de desarrollo	**Prioridad: 520**
Entrada	OG09 Entregable final	
Definición	Tras haber realizados todas las pruebas con el usuario, se genera un ejecutable que será subido a una plataforma de distribución de manera gratuita.	
OE32 Android Application Package (apk) de Android		
Roles	Equipo de desarrollo	**Prioridad: 530**

Entrada	OG09 Entregable final
Definición	Luego de todas las pruebas con el dispositivo móvil, se procede a generar un Android Application Package (apk) que será subido a una plataforma de distribución con ningún costo.

3.3 Elaboración

3.3.1 Tareas

Tabla 5. *Configuración Unity*

Tarea	OE01 Configuración Unity	
Roles	Equipo de desarrollo	**Prioridad: 150**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el software de Unity con licencia gratuita para ubicar en entorno 3D, creando un proyecto con estructura para poder realizar la integración con otros programas.	

Ejecución Característica

Primero se instaló Unity en el computador, se registra en la página oficial y se selecciona la licencia gratuita que nos va a funcionar para este proyecto, Luego de haber instalado correctamente creamos un nuevo proyecto, por defecto Unity no nos genera una estructura en el proyecto por lo que empezaremos creando la estructura por buenas prácticas, se generan las carpetas y relaciones necesarias para el proyecto.

Verificación

El programa funciona correctamente y la estructura esta creada en base a las necesidades

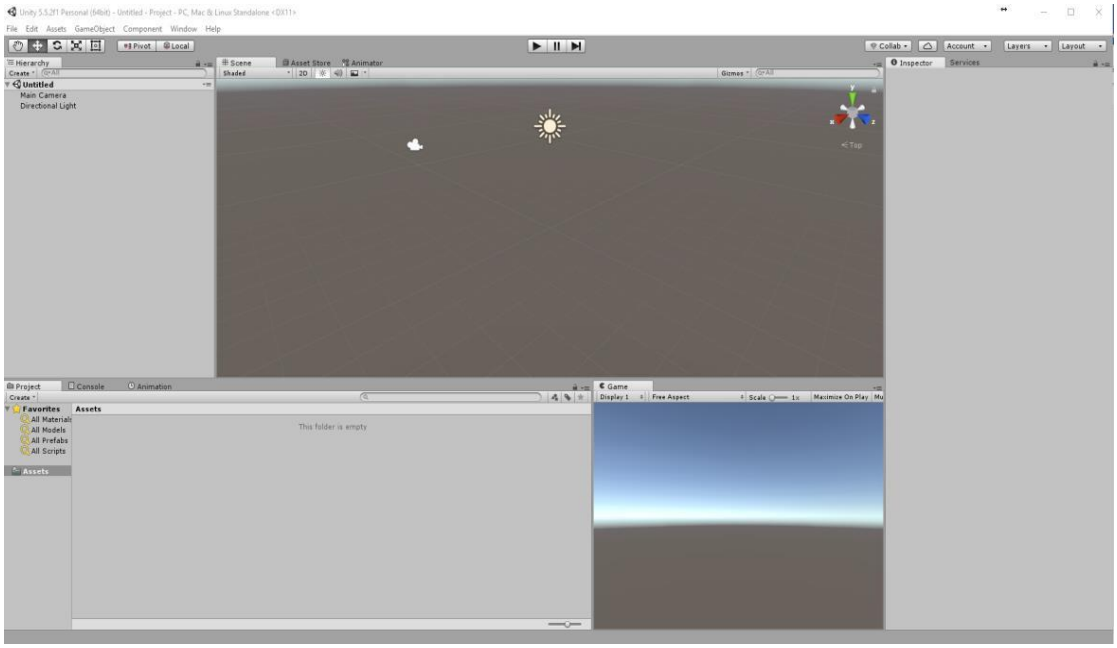

Figura 17. Configuración Unity

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 6.*Configuración Orion*

Tarea	OE02 Configuración Orion	
Roles	Equipo de desarrollo	**Prioridad: 160**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el software de Orion para poder utilizar una implementación con Leap Motion y poder calibrar el dispositivo.	

Ejecución Característica

La instalación de Orion es muy importante ya que nos permite el uso del Leap Motion con Oculus Rift y también nos permite calibrar de mejor manera el Leap Motion, por esta razón se lo instala previamente, luego de instalarlo se inicia el servicio por el que va a funcionar el software.

Verificación

El programa está corriendo en background.

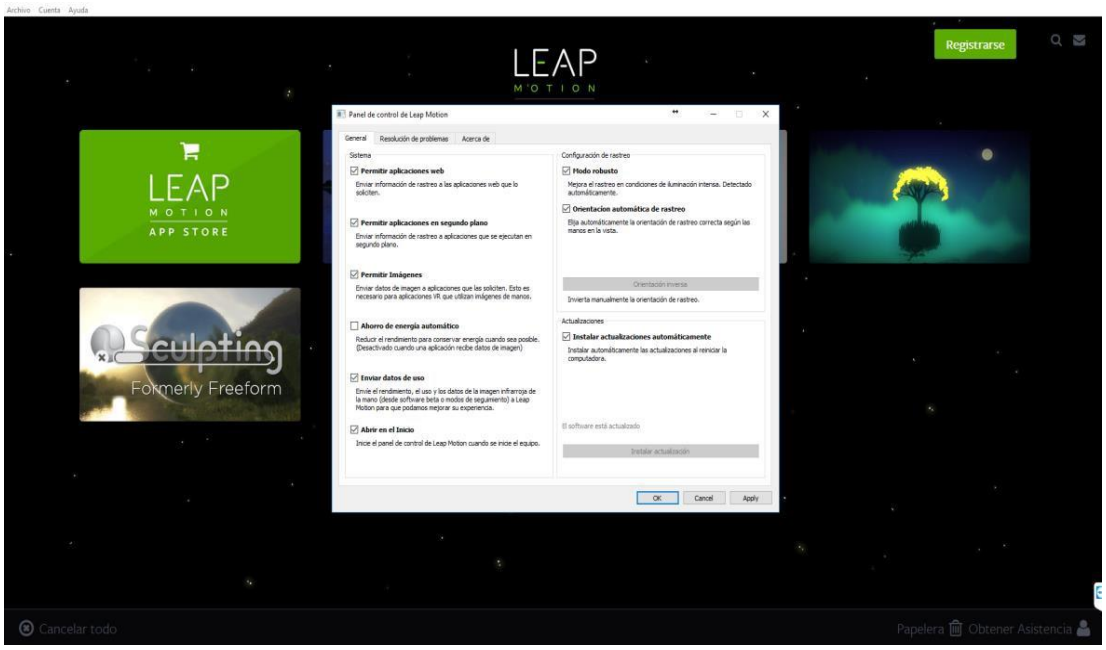

Figura 18. Instalación Leap Motion

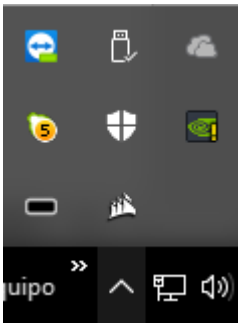

Figura 19. Servicio Leap Motion funcionando

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 7. *Configuración Cinema 4D*

Tarea	OE03 Configuración Cinema4D	
Roles	Equipo de desarrollo	**Prioridad: 200**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el software funcionando en el computador donde se realizarán los diseños de la simulación.	

Ejecución

Característica

Se registra en la página oficial donde nos permite comprar una licencia de uso personal, luego de descargar instalamos el programa, y lo abrimos para poder empezar a utilizarlo.

Verificación

El programa está funcionando correctamente.

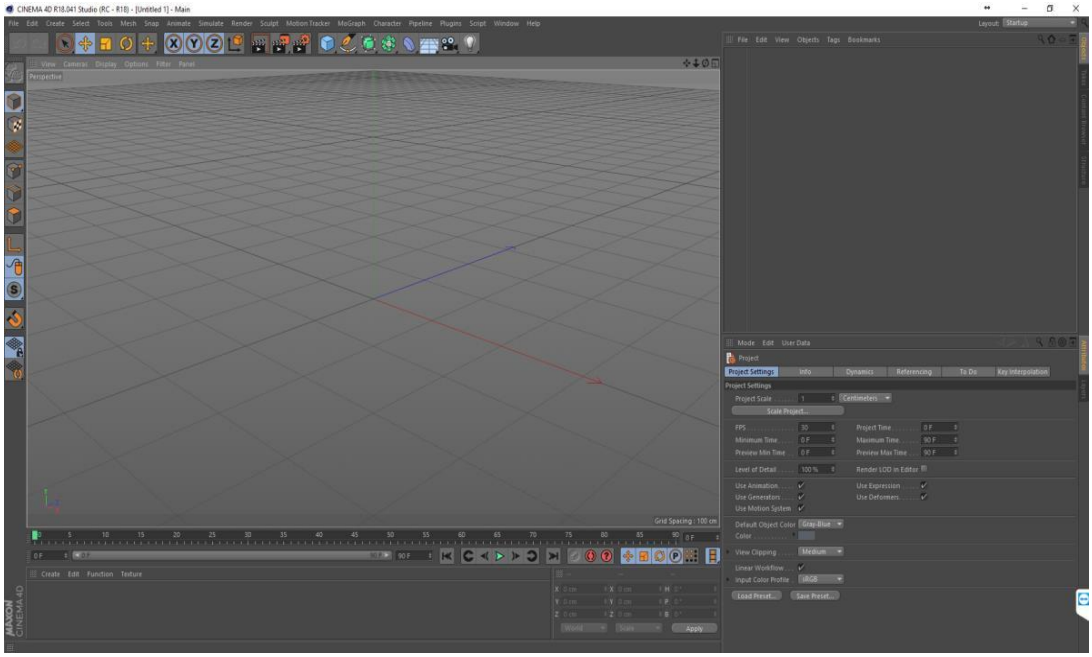

Figura 20. Instalación Cinema4D

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 8.*Configuración Visual Studio*

Tarea	OE04 Configuración Visual Studio	
Roles	Equipo de desarrollo	**Prioridad: 170**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se configura el Visual Studio para que se integre con Unity, y de esta manera poder utilizarlo directamente como complemento con el código de la simulación desde Unity.	

Ejecución Característica

Se realiza la instalación y la integración del Visual Studio con el Unity para que funcione el autocompletado y otras opciones que ayudan en el desarrollo del proyecto, se entra en el Sync MonoDevelop Project para cambiar el uso al Visual Studio, y se descargaran los complementos necesarios del Visual Studio para funcionar con Unity.

Verificación

Visual Studio funciona correctamente con Unity.

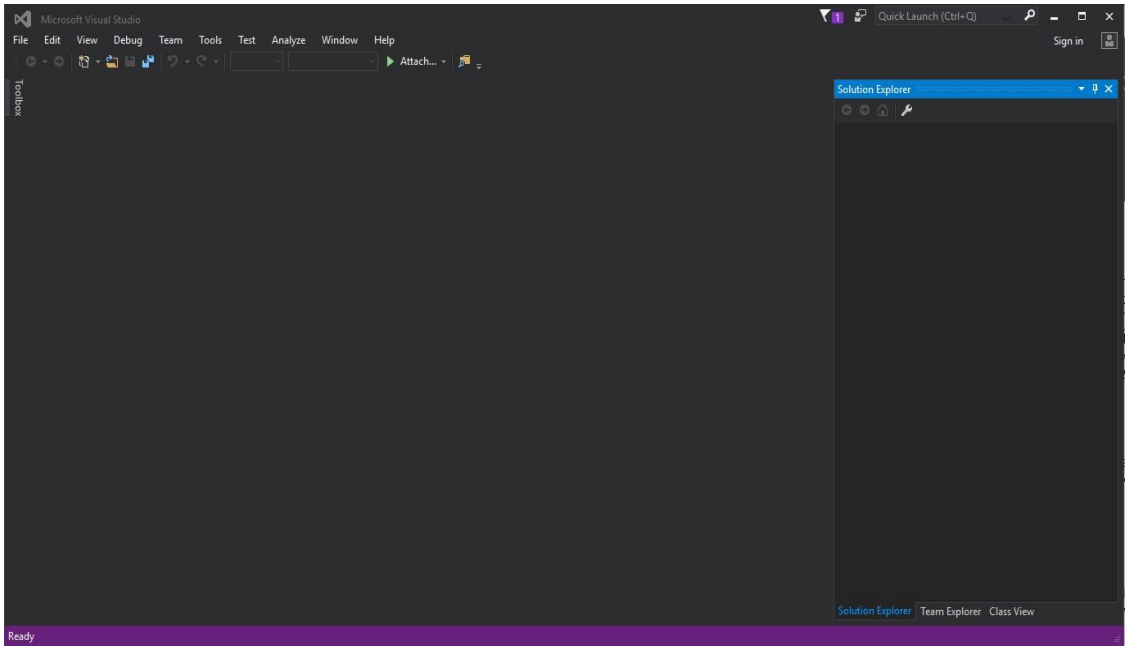

Figura 21. Configuración Visual Studio

Manejo de Problemas

Hubo un problema al bajarse los complementos de la versión equivocada por lo que no funciono la integración, descargando los complementos de la versión correcta se corrigió el error**.**

Tabla 9.*Configuración Magica Voxel*

Tarea	OE05 Configuración Magica Voxel y Vuforia	
Roles	Equipo de desarrollo	**Prioridad: 180**
Entrada	OG01 configuración del ambiente de desarrollo	
Definición	Se Realiza el registro en el portal de Vuforia con licencia gratuita y se configura Magica Voxel para poder modelar	

Ejecución Característica

Se descarga el ejecutable de Magica Voxel y se lo poner a funcionar, se registra en la página de Vuforia, y se genera una clave con licencia de estudiante, inicializamos un proyecto y creamos la base de datos principal.

Verificación

Los programas están funcionando correctamente.

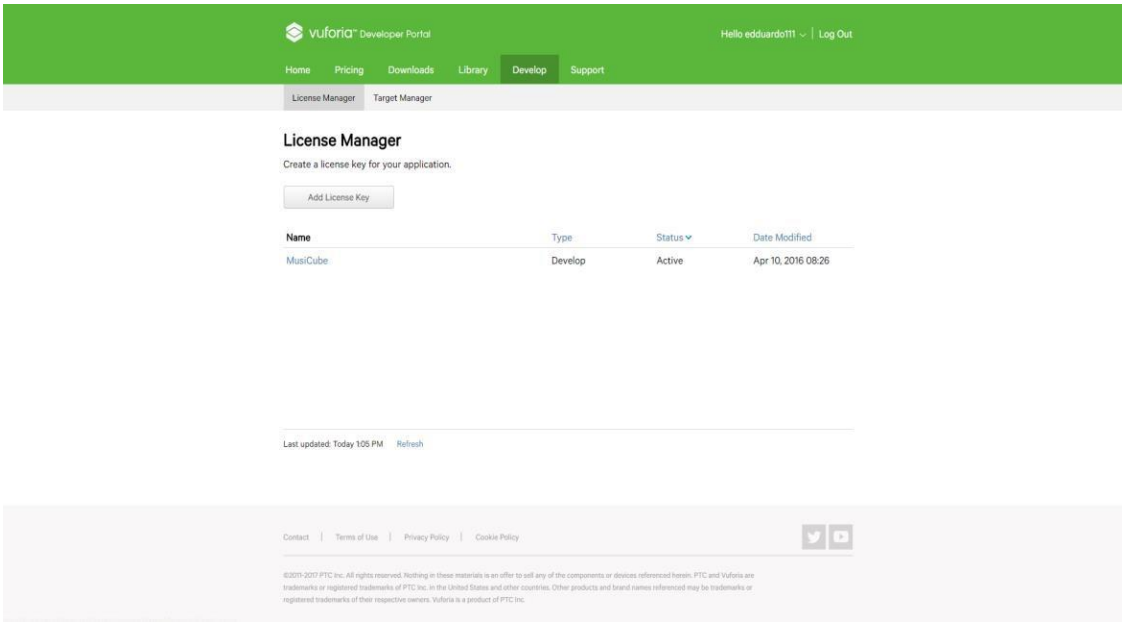

Figura 22. Vuforia con Licencia Configurada

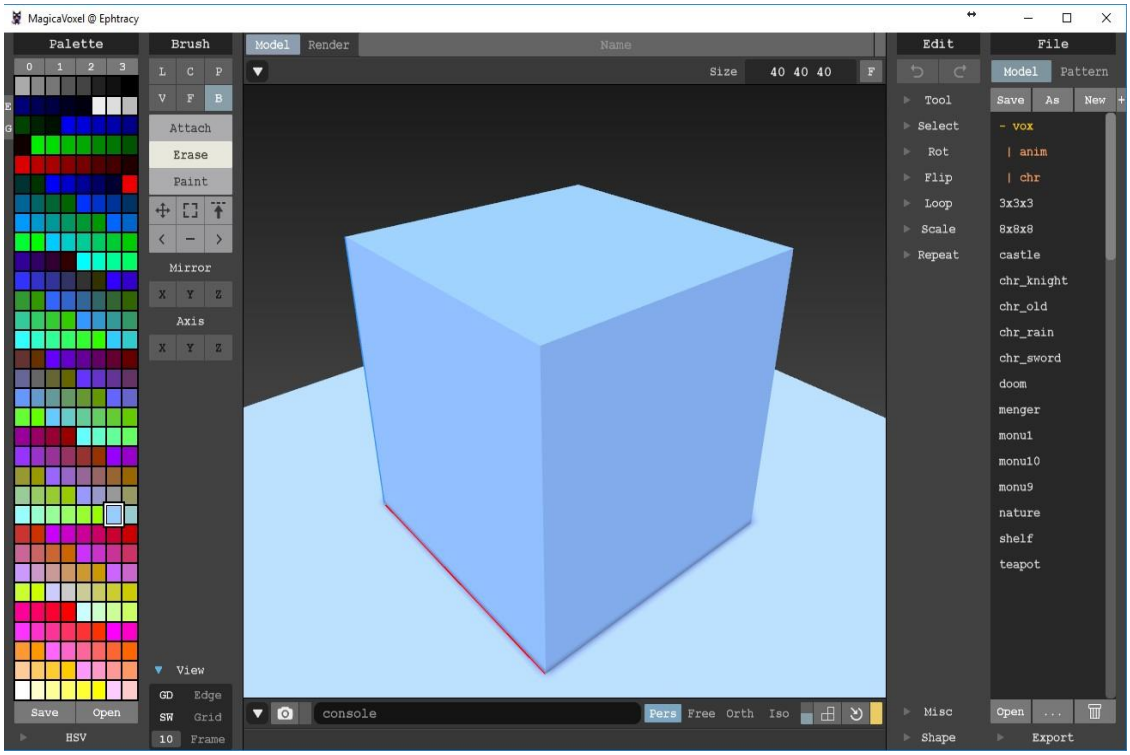

Figura 23. Magica Voxel en funcionamiento

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 10.*Configuración Oculus Rift*

Tarea	OE06 Configuración Oculus Rift	
Roles	Equipo de desarrollo	**Prioridad: 210**
Entrada	OG02 configuración de periféricos	
Definición	Se configura el Oculus Rift y se calibra para poder funcionar en el computador.	

Ejecución Característica

Se descarga el instalador del Oculus y se procede a instalarlo, se tiene que ubicar la dirección correcta de los conectores que se van a utilizar y calibrar correctamente con la persona y el entorno en el que se lo va a utilizar.

Verificación

El menú de Oculus está funcionando correctamente sin errores.

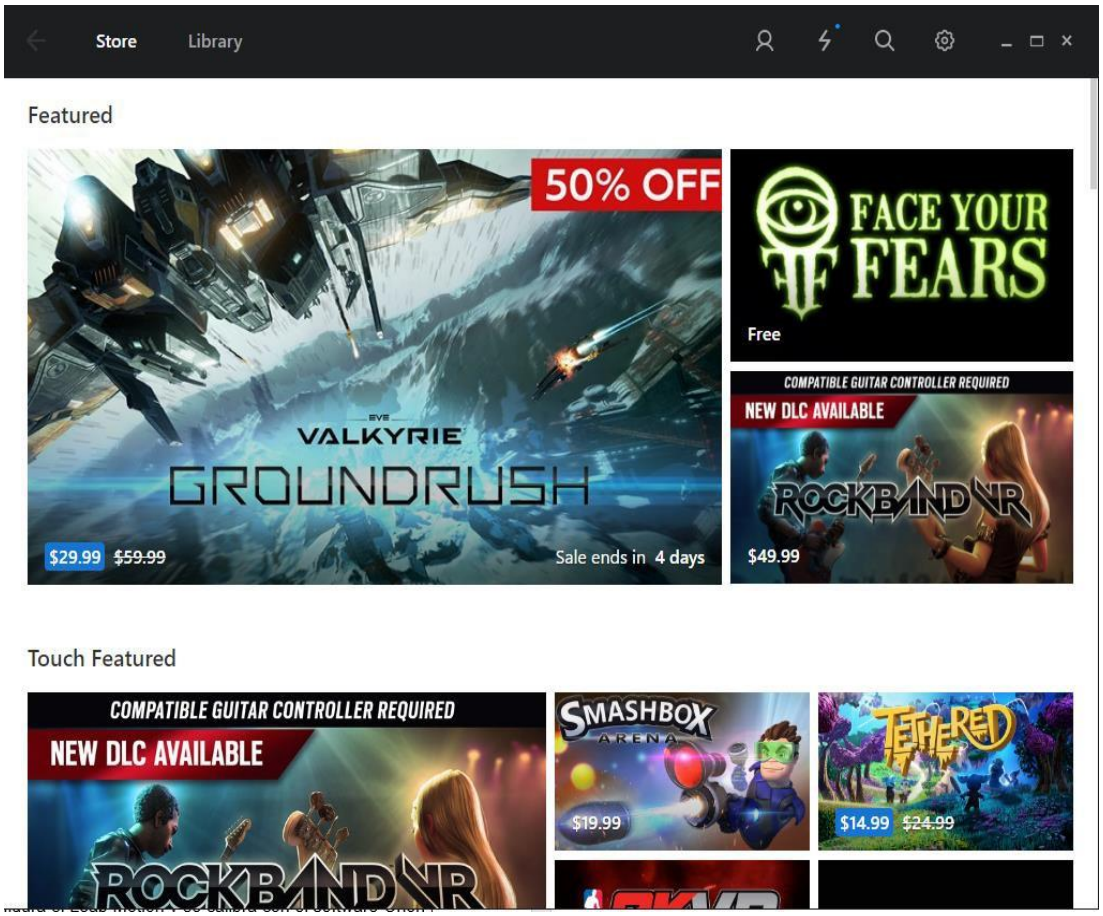

Figura 24. Menú Oculus Rift

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 11.*Configuración Leap Motion*

Tarea	OE07 Configuración Leap Motion	
Roles	Equipo de desarrollo	**Prioridad:**
		220
Entrada	OG02 configuración de periféricos	
Definición	Se configura el Leap Motion y se calibra con el software Orion previamente instalado para poder funcionar en conjunto con el Oculus Rift.	

Ejecución Característica

Se configura el Leap Motion utilizando el software de Orion, que nos permite hacer la integración con el Oculus ya configurado, para poder calibrar se utiliza un espejo para reflejar el sensor infrarrojo y que el dispositivo pueda calcular la distancia correctamente, luego se ubica en el modelo de prueba para comprobar que el funcionamiento es correcto.

Verificación

El demo de Leap Motion funciona correctamente.

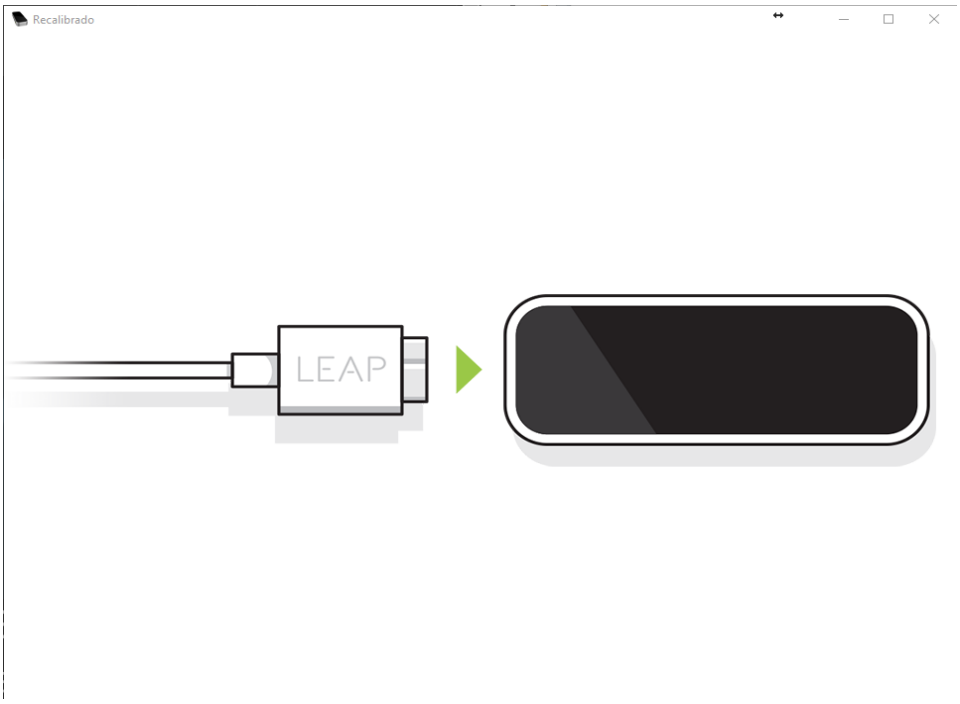

Figura 25. Ventana de Leap Motion Funcionando

Manejo de Problemas

Al momento de calibrar también se puede utilizar un vidrio, pero el reflejo que dio en el día no fue suficiente, se perdía algunos datos de reflejo por lo tanto la calibración fallo, y se procedió a calibrar con espejo**.**

Tabla 12.*Configuración de funcionamiento de periféricos en Unity*

Tarea	OE08 Configuración de funcionamiento de periféricos en Unity	
Roles	Equipo de desarrollo	**Prioridad: 230**
Entrada	OG02 configuración de periféricos	
Definición	Se configura la integración entre Oculus Rift y Leap Motion como un modelo prefab en Unity y poder utilizarlo en la simulación.	

Ejecución Característica

En el proyecto ya creado de Unity se procedió a crear el prefab de la integración entre los dos dispositivos en la escena de la simulación, luego se vinculó con los scripts necesarios extraídos de la librería de Oculus Rift y se utilizó el editor para probar el funcionamiento de los periféricos.

Verificación

El editor no muestra ningún error en consola y el prefab está funcionando.

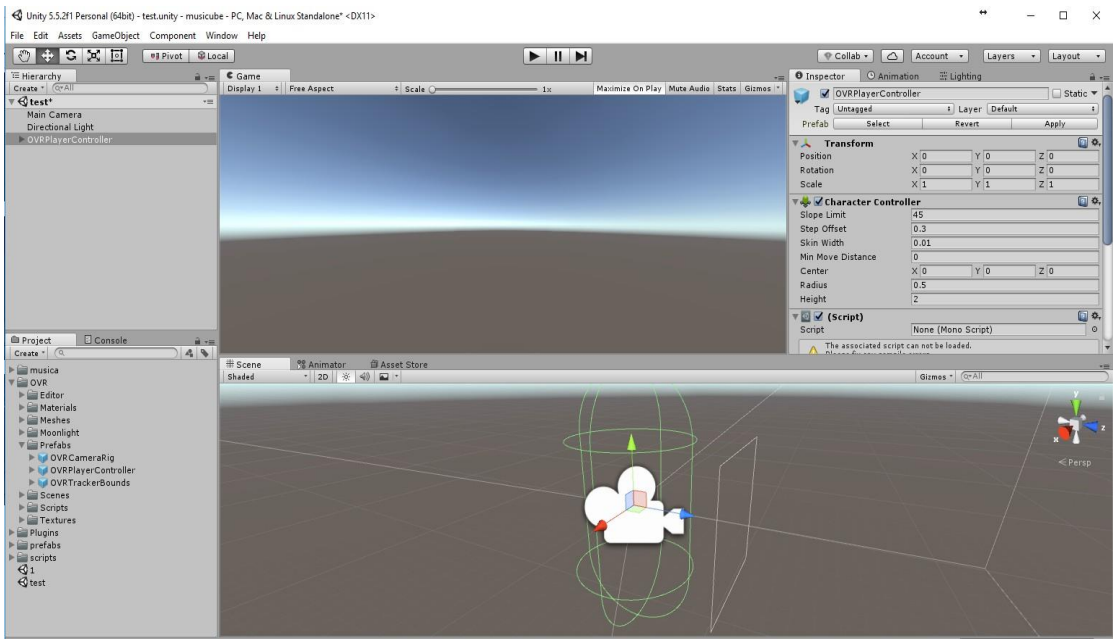

Figura 26. Modulo Oculus Rift en Unity

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 13. *Diseño de la plataforma de la simulación*

Tarea	OE09 Diseño de la plataforma de la simulación	
Roles	Equipo de desarrollo	**Prioridad:**
		250
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña un modelo de la plataforma por la que se va a interactuar en la simulación desde Magica Voxel.	

Ejecución Característica

En la simulación se debe hacer lo mejor posible para que sea realista, como el usuario no se puede mover se optó por realizar una plataforma que será la que interactúe con el entorno y llevara al usuario a través de la simulación, se utilizó mágica voxel para realizar este modelo en gráficos tipo voxel, ya que el entorno entero será en voxel, luego se lo importa en cinema 4D para mejorar el rendimiento y por último se lo importa en Unity, se le acomoda y combina con el prefab de Oculus Rift para poder utilizar la plataforma.

Verificación

La plataforma se puede observar mediante el Oculus Rift de la manera correcta.

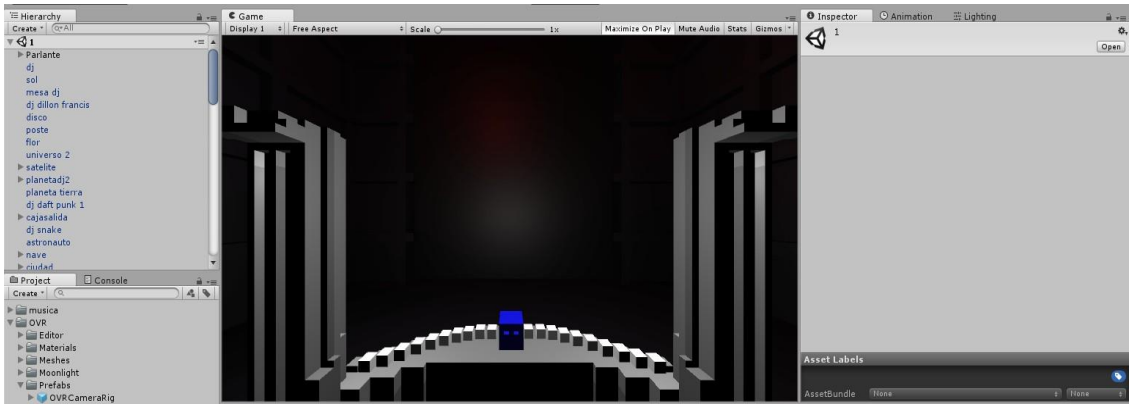

Figura 27. Oculus Rift en funcionamiento desde Unity

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 14.*Diseño del entorno de la simulación*

Tarea	OE10 Diseño del entorno de la simulación	
Roles	Equipo de desarrollo	**Prioridad: 260**
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña el entorno por partes en mágica voxel, se lo une y se lo optimiza en Cinema4D para poder importarlo en Unity	

Ejecución Característica

En la simulación es muy importante el entorno que se va a observar, este entorno se dividió en tres partes, al querer hacer futurista se utiliza el primer entorno como una salida desde el espacio, en el segundo entorno es un planeta de más cerca, y en el tercer entorno es en una ciudad dentro de ese planeta, los tres entorno tendrán apariencia tipo voxel, pero en este caso los tres son desarrollado en cinema 4D debido a el tamaño del modelo, ya que mágica voxel solo nos permite hacer modelos pequeños, aunque se procede a hacer con mágica voxel modelos más pequeños como detalles en los edificios o en el espacio, y por último se lo importa en Unity (Watkins, 2012, p. 72).

Verificación

El entorno es renderizado correctamente.

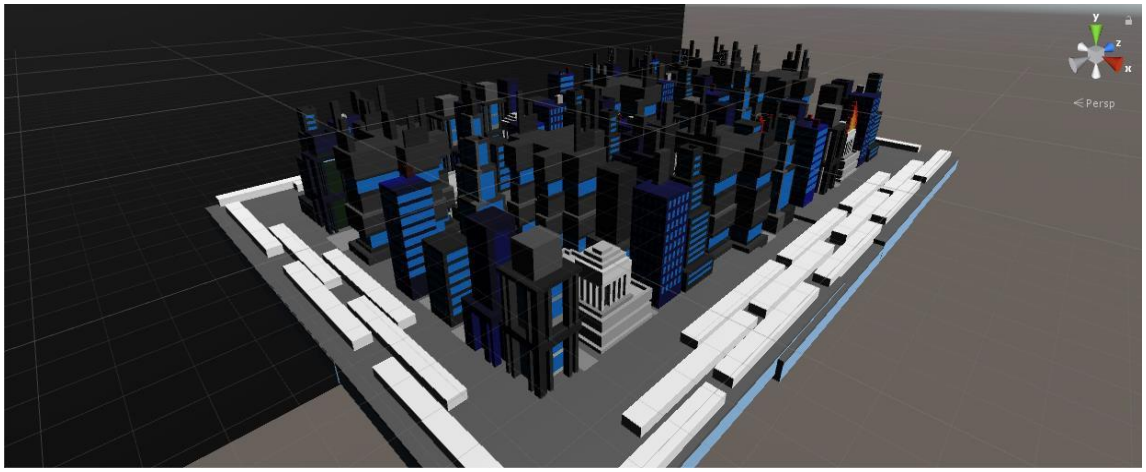

Figura 28. Entorno Voxel renderizado

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 15.*Diseño de personajes*

Tarea	OE11 Diseño de personajes	
Roles	Equipo de desarrollo	**Prioridad: 270**
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña los personajes principales y secundarios que se podrán observar en la simulación.	

Ejecución Característica

La simulación contiene personajes, estos personajes representan a Djs famosos de la época actual, al ser una simulación basada en música, estos modelos son atractivos al usuario con apariencia de personas, pero sin perder el tipo voxel.

Verificación

Los modelos de los personajes se renderizan correctamente.

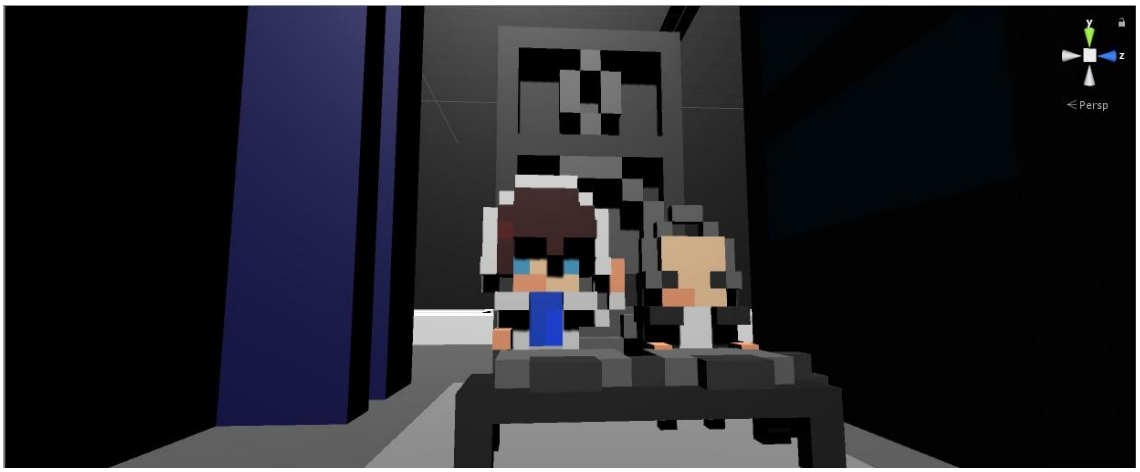

Figura 29. Modelos de personajes

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 16.*Diseño de modelos dinámicos*

Tarea	OE12 Diseño de modelos dinámicos	
Roles	Equipo de desarrollo	**Prioridad: 280**
Entrada	OG03 Diseño del ambiente	
Definición	Se diseña los modelos que van a ser dinámicos, con el complemento de script y superficies en trigger que funcionaran con el script de reconocimiento.	

Ejecución Característica

El resto de modelos se caracterizan por ser estáticos, significa que no pueden tener ningún tipo de articulación o interacción ya que se encuentran con propiedades de colisión y de ancla, por lo tanto habrán modelos que tienen que interactuar todo el tiempo, estos se lo desarrolla por partes estáticamente en mágica voxel y se los une en cinema 4D para que sean dinámicos, por ultimo al importar en Unity se los poner con propiedad de trigger para que puedan cambiar sus propiedades físicas fácilmente.

Verificación

El modelo dinámico está configurado correctamente.

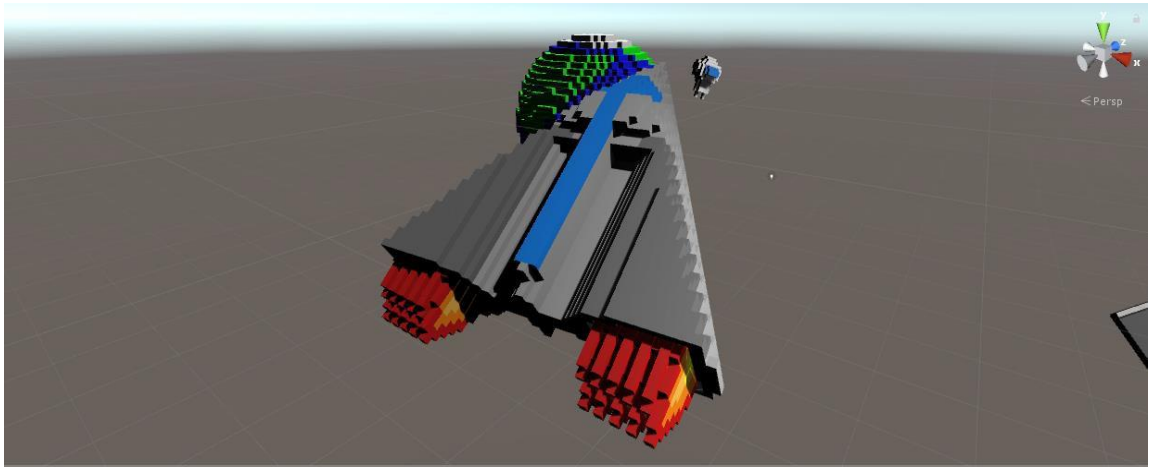

Figura 30. Modelo dinámico

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 17.*Animaciones primarias y secundarias*

Tarea	OE12 Animaciones primarias y secundarias	
Roles	Equipo de desarrollo	**Prioridad: 290**
Entrada	OG03 Diseño del ambiente	
Definición	Se realizan las animaciones en los modelos primarios y secundarios que no van a reaccionar con el script.	

Ejecución Característica

Los modelos estáticos aunque no pueden tener movimiento independiente pueden tener movimiento del modelo completo, a esto se le llama animación secundaria, por otro lado los modelos dinámicos pueden tener todo tipo de movimiento configurado, estos tienen animaciones primarias, para lograr esto utilizamos el animator de Unity y animar todos los modelos posibles de la simulación para que sea más atractivo al usuario y no tan estático, cabe recalcar que los modelos dinámicos tendrán otro tipo de movimientos que no serán controlados por animator sino por scripts más adelante.

Verificación

Las animaciones funcionan correctamente.

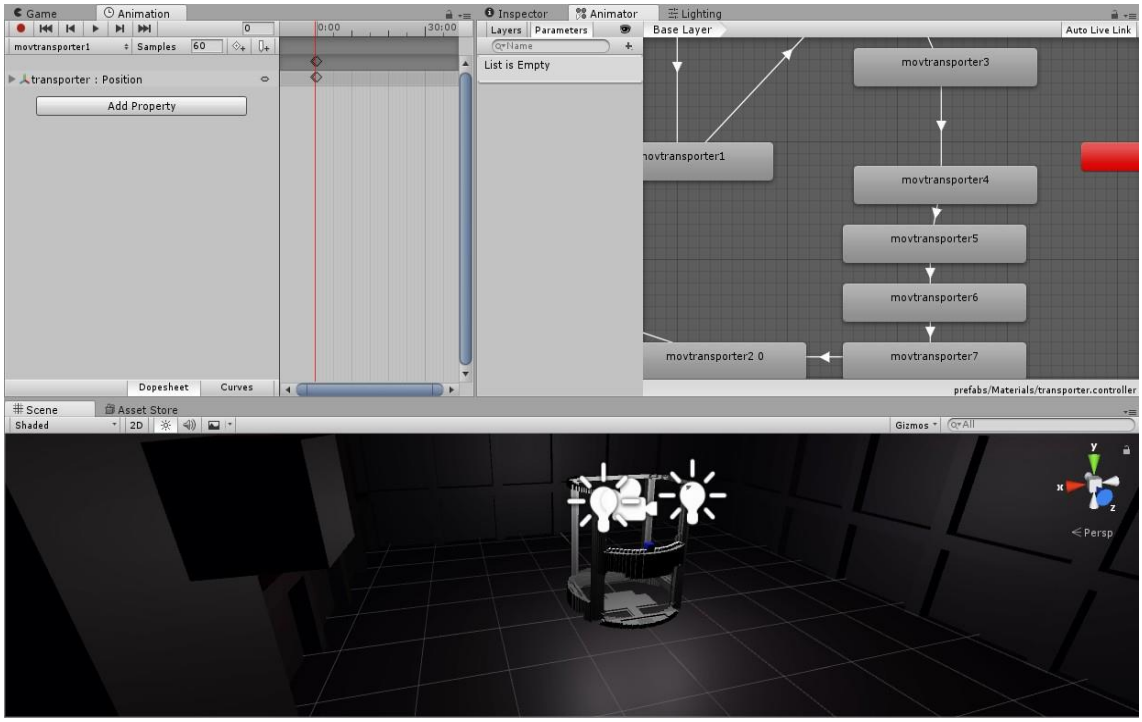

Figura 31. Controlador de animaciones

Manejo de Problemas

Cuando un modelo tiene una relación padre e hijo, y se anima primero el hijo, causa problemas en las propiedades de ubicación en los 3 ejes por lo que la animación puede fallar, esto se puede controlar, se procedió a quitar la relación, tras animar por separado unir otra vez la relación y comprobar que funcione, caso contrario se tendría que animar solo el modelo padre al no ser modelos dinámicos**.**

Tabla 18. *Iluminación de la simulación*

Tarea	OE14 Iluminación de la simulación	
Roles	Equipo de desarrollo	**Prioridad: 295**
Entrada	OG03 Diseño del ambiente	
Definición	Ya que al inicio de la simulación se encuentra en el espacio se desarrolla la contra iluminación, pero en el tercer escenario se encuentra en la ciudad donde necesita otra iluminación.	

Ejecución Característica

La simulación cuenta con tres entornos los cuales tienen que ser correctamente iluminados, en el primer y segundo entorno se encuentra en el espacio por lo que la iluminación tiene que ser por punto para que se pueda observar, pero todo el alrededor sea obscuro, en el tercer entorno, se encuentra en una ciudad por lo que la iluminación es direccional y de ambiente para que se pueda observar como realmente se podría.

Verificación

La iluminación se visualiza correctamente.

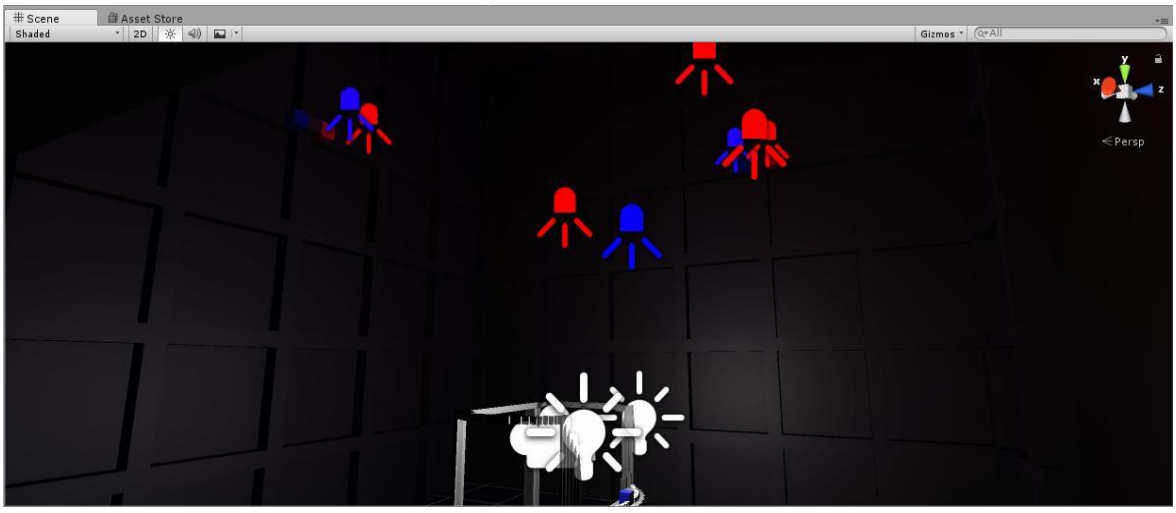

Figura 32. Módulo de iluminación

Manejo de Problemas

Al estar los tres entornos en la misma escena la iluminación puede

cruzarse y dañar el resto de entornos, para esto se ubicó un separador entre los entornos para que cada uno actúe independientemente, aunque los tres se encuentren en la misma escena**.**

Tabla 19. *Desarrollo de script para reconocer frecuencias a tiempo real*

Tarea	OE15 Desarrollo de script para reconocer frecuencias a tiempo real	
Roles	Equipo de desarrollo	**Prioridad: 300**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla en lenguaje de programación C# el reconocimiento de frecuencias utilizando la librería spectrum.	

Ejecución Característica

En Unity creamos en la carpeta destinada para los scripts el que va a ser reconocimiento de frecuencias, con la integración con Visual Studio podremos importar la librería de spectrum que nos permite identificar los valores de la frecuencia de un sonido, luego clasificamos e interpretamos estos datos a las frecuencias de las notas musicales, para proceder a clasificarlas e identificar bajos, agudos y medios de una canción (Creative, 2017).

Verificación

El script está funcionando parcialmente hasta que se le implemente una fuente de audio.

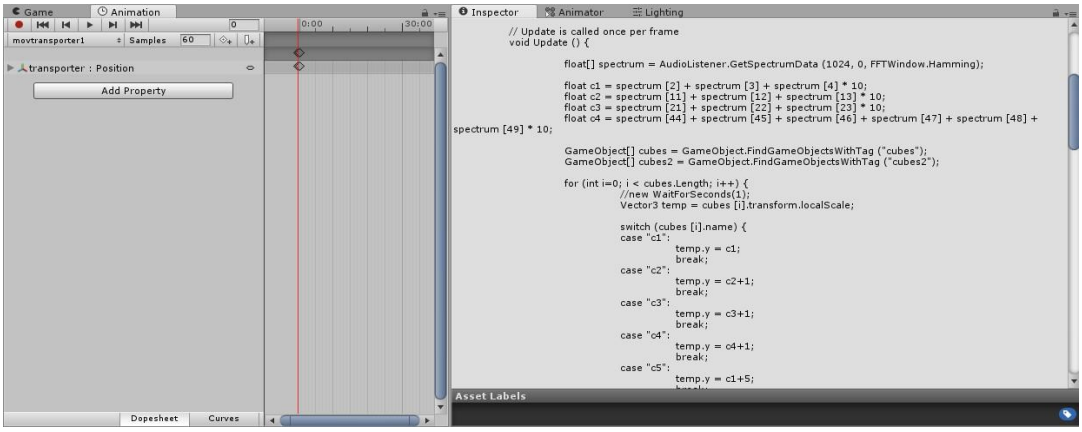

Figura 33. Script de reconocimiento de frecuencias

Manejo de Problemas

El rango de frecuencias depende de la entrada de audio, si este rango es mal calibrado, no se notará el efecto o el efecto estará descuadrado, utilizando un afinador de guitarra se puede calcular los valores delas frecuencias para estar ajustados en los rangos necesarios**.**

Tabla 20. *Desarrollo de script para modificar propiedades de modelos*

Tarea	OE16 Desarrollo de script para modificar propiedades de modelos	
Roles	Equipo de desarrollo	**Prioridad: 310**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla en lenguaje de programación C# el comportamiento de los modelos dinámicos utilizando las informaciones obtenidas del script de frecuencias.	

Ejecución Característica

Ahora que se realizó el script de reconocimiento se lo utilizo en variables que se encontraran actualizándose sesenta veces por segundo y asignarles estas a los valores que queremos modificar de los modelos dinámicos, para que estos reaccionen a las frecuencias musicales.

Verificación

Las propiedades de los modelos dinámicos están cambiando de acuerdo a las variables utilizadas.

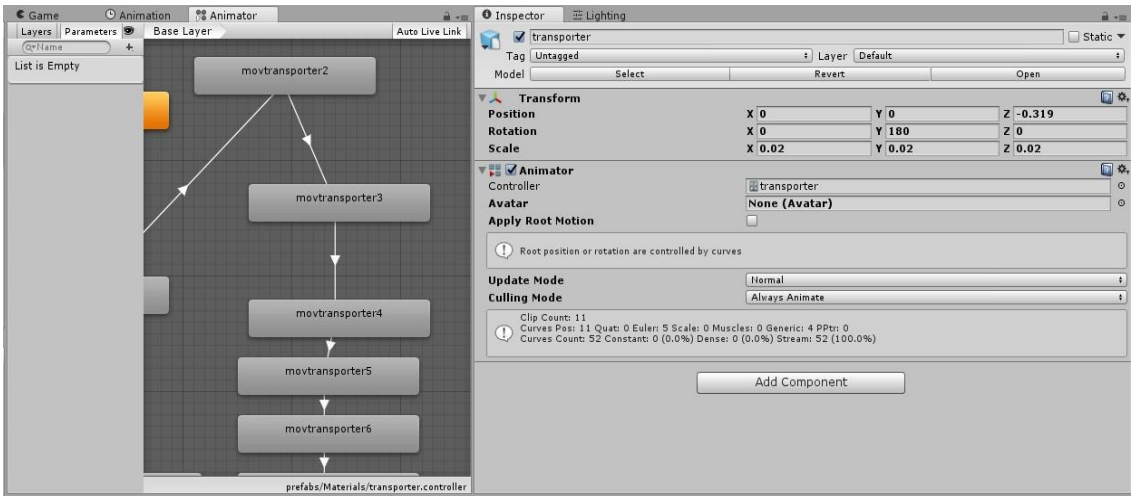

Figura 34. Controlador de animaciones dinámico

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 21.

Desarrollo de script para reconocimiento de música por AUX

Tarea	OE17 Desarrollo de script para reconocimiento de música por AUX	
Roles	Equipo de desarrollo	**Prioridad: 320**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla en lenguaje de programación C# el reconocimiento del dispositivo que va a reproducir la canción mediante el cable AUX.	

Ejecución Característica

En la simulación se tomará como entrada de sonido al canal AUX, para que pueda conectar cualquier reproductor de música, esto se lo realiza mediante un script que vincula directamente la tarjeta de sonido para poder recibir las frecuencias de la música a tiempo real en la simulación.

Verificación

Se puede escuchar dentro de la simulación la entrada auxiliar.

Manejo de Problemas

Si el módulo del micrófono no es correctamente redactado con el nombre original de la tarjeta de sonido, el script no tendrá efecto por lo que hay que entrar en la configuración de la tarjeta de sonido primero para comprobar el nombre del módulo**.**

Tabla 22.*Desarrollo complemento Audio Source*

Tarea	OE18 Desarrollo complemento Audio Source	
Roles	Equipo de desarrollo	**Prioridad: 330**
Entrada	OG04 Reconocimiento de frecuencias	
Definición	Se desarrolla el complemento del audio source con el script de frecuencias, y para reproducir la música utilizando el script de reconocimiento AUX.	

Ejecución Característica

Ahora que se realizó todos los scripts y vinculaciones necesarias, podemos vincular todo a un complemento Audio Source de Unity que nos permite utilizar la música que viene de entrada (script periférico) y poder identificar y reproducir las frecuencias dentro de la simulación (script frecuencias), para poder calcular las frecuencias sesenta veces por segundo y poder tener valores en tiempo real, este audio source se conecta directamente con todos los modelos dinámicos por lo que estos reaccionaran a la música que viene de entrada.

Verificación

Se puede visualizar el comportamiento de los modelos dinámicos con la música de entrada AUX.

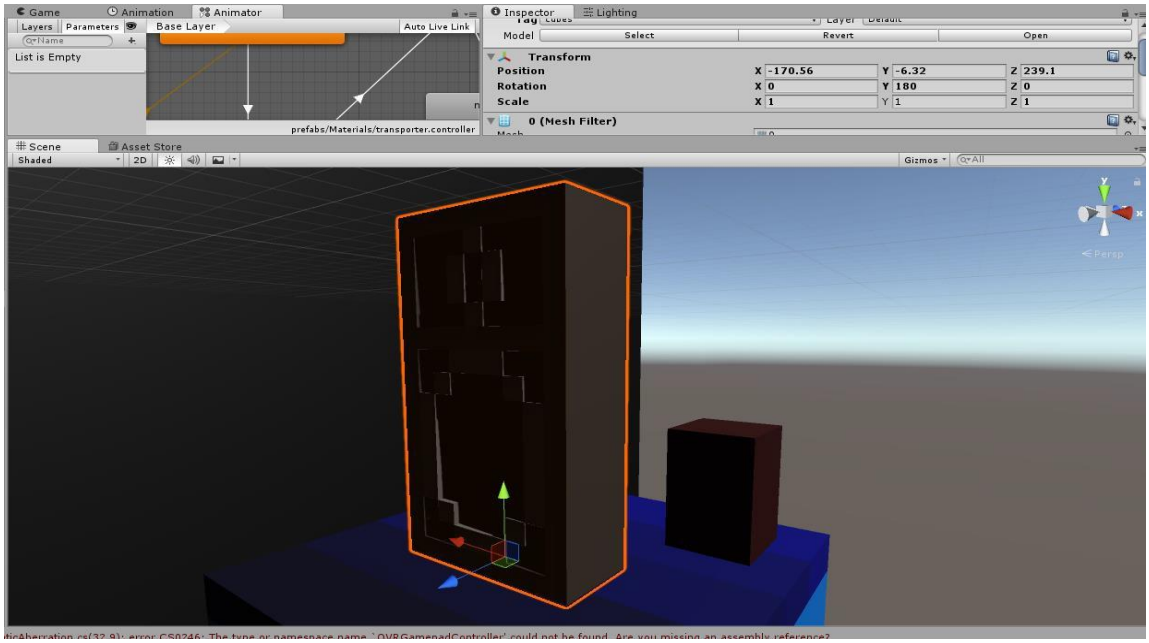

Figura 35. Modelo dinámico funcionando con script

Manejo de Problemas

El complemento tiene que ser ubicado en algún objeto 3D para poder funcionar ya que se comporta de esa manera, aunque no es renderizado como modelo 3D si se comporta como uno para poder funcionar dentro de la escena**.**

Tabla 23.*Administración de colisiones*

Tarea	OE19 Administración de colisiones	
Roles	Equipo de desarrollo	**Prioridad: 350**
Entrada	OG05 Gameplay	
Definición	Debido al dispositivo de Leap Motion hay modelos que pueden ser tocados y que interactúan con el entorno, esto significa que algunos modelos necesitaran utilizar propiedades físicas que les permitan realizar estas acciones.	

Ejecución Característica

En Unity se utiliza el complemento de rigid body para hacer esto, ubicando dentro de algunos objetos esta propiedad, se comportarán con propiedades físicas, y esto permite la interacción con el modelo de las manos que tiene colisión, luego por último se administró cual será trigger y que reaccionará con la función OnTriggerEnter().

Verificación

Se pueden colisionar los objetos.

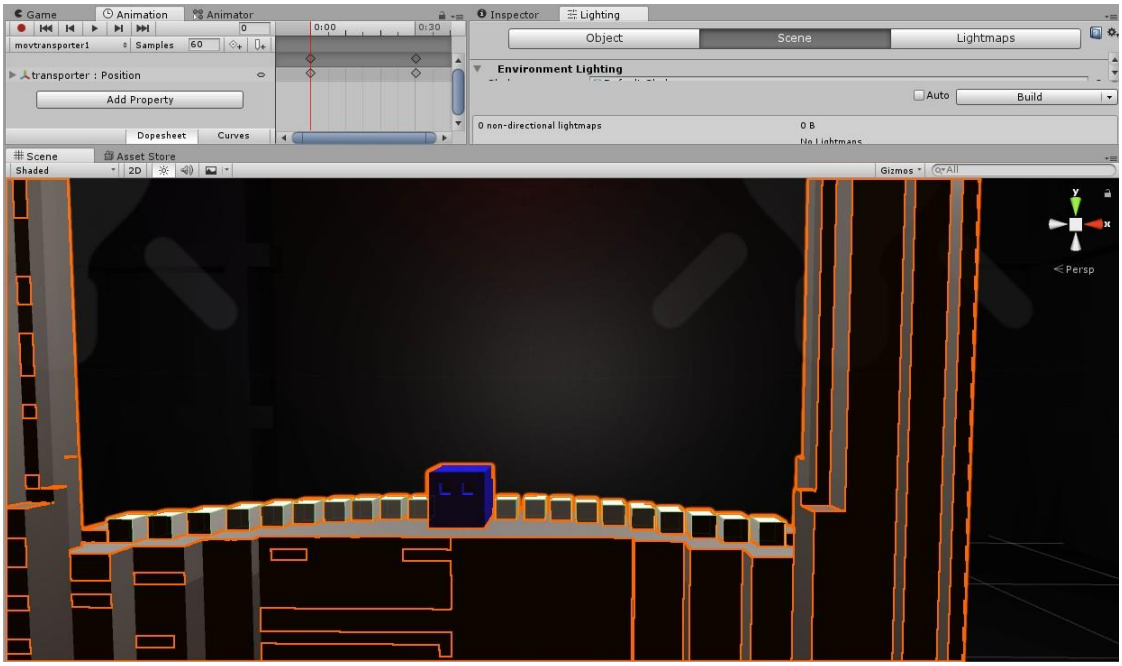

Figura 36. Colisión de objetos

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 24. *Administración de triggers*

Tarea	OE20 Administración de triggers	
Roles	Equipo de desarrollo	**Prioridad: 360**
Entrada	OG05 Gameplay	
Definición	Hay modelos del entorno que no necesitan ningún tipo de interacción o solo necesitan interacción de control y validación, por lo tanto, se les aplica propiedades físicas de triggers.	

Ejecución Característica

En Unity se utiliza el complemento de rigid body para hacer que utilice la función del trigger, que permite mediante funciones controlar eventos al momento de validar si el objeto ha sido tocado por el usuario o no.

Verificación

Se pueden colisionar los objetos.

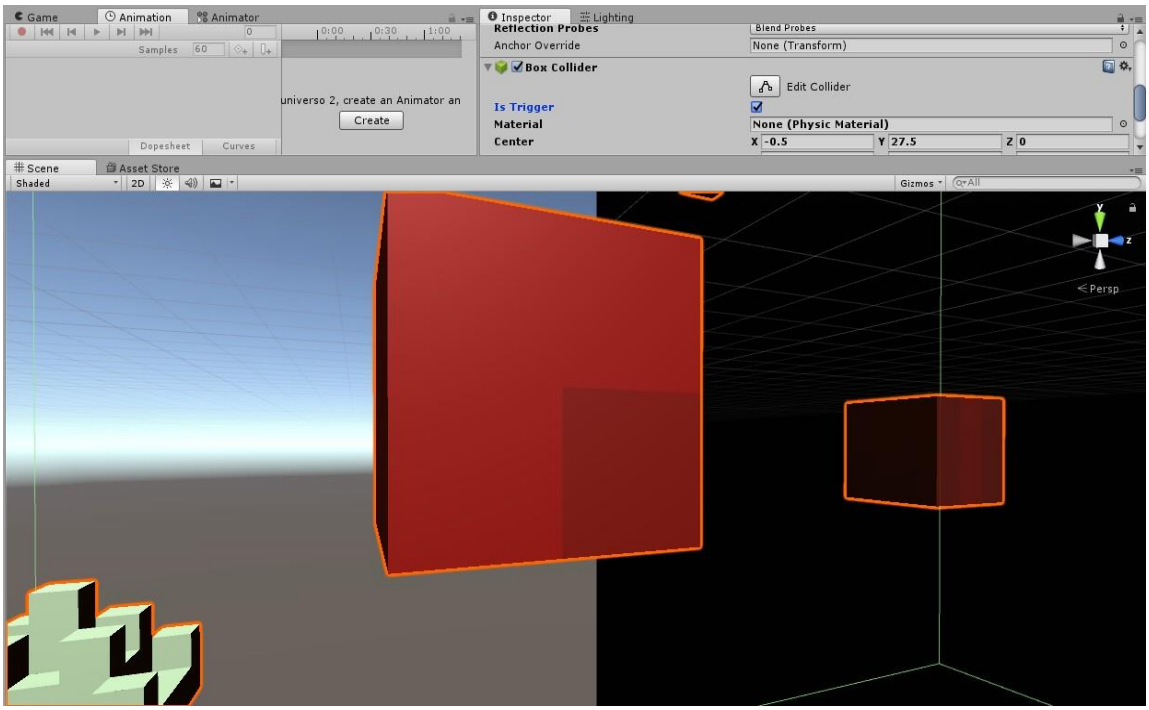

Figura 37. Objeto con trigger

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 25. *Desarrollo interfaces realidad virtual*

Tarea	OE21 Desarrollo interfaces realidad virtual	
Roles	Equipo de desarrollo	**Prioridad: 400**
Entrada	OG06 Desarrollo de interfaces	
Definición	En las interfaces de realidad virtual tienen que tener interacción amigable con el usuario, estas interfaces tienen que tener los controladores simples.	

Ejecución Característica

Utilizando el complemento de canvas de Unity, acoplamos el anchor para que los paneles se observen en el Oculus Rift, podemos luego poner los controladores necesarios en cada uno de los paneles para desarrollar las interfaces, y por último darles un diseño simple.

Verificación

Se pueden observar las interfaces correctamente.

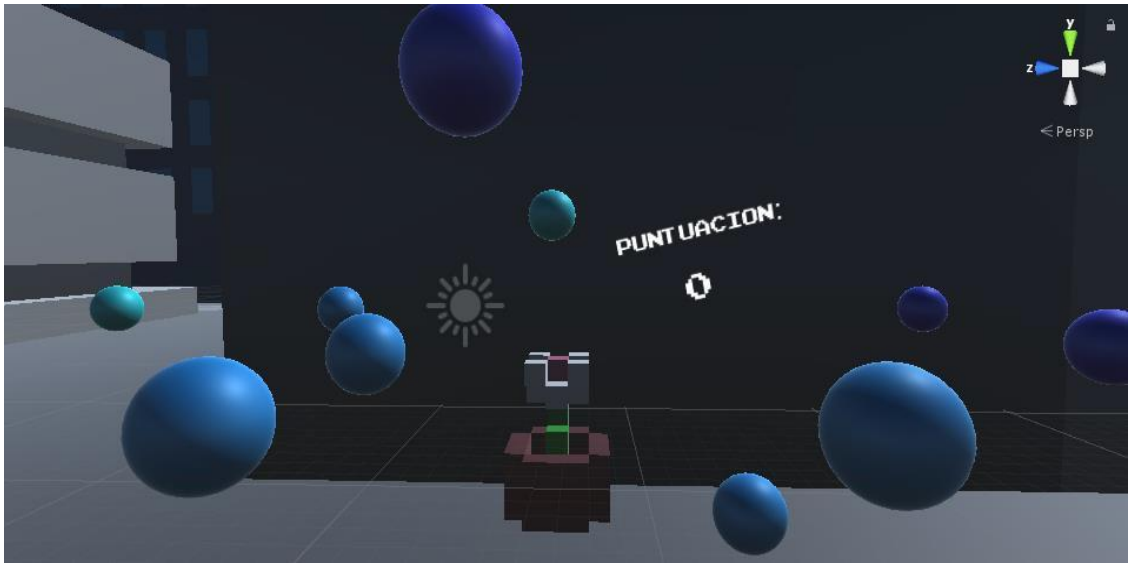

Figura 38. Interfaz con Oculus Rift

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 26. *Desarrollo interfaces dispositivo móvil*

Tarea	OE22 Desarrollo interfaces dispositivo móvil	
Roles	Equipo de desarrollo	**Prioridad: 410**
Entrada	OG06 Desarrollo de interfaces	
Definición	Se desarrollan las interfaces para dispositivos móviles, con las propiedades correctas para que sean responsive a cualquier dispositivo.	

Ejecución Característica

Utilizando el complemento de canvas de Unity, utilizamos los anchors para acomodar los paneles, estos paneles se acomodan a la pantalla de celular o tablet sin importar la resolución ya que se autoajustará a este, de esta manera se convierten en responsive.

Verificación

Se pueden observar las interfaces correctamente.

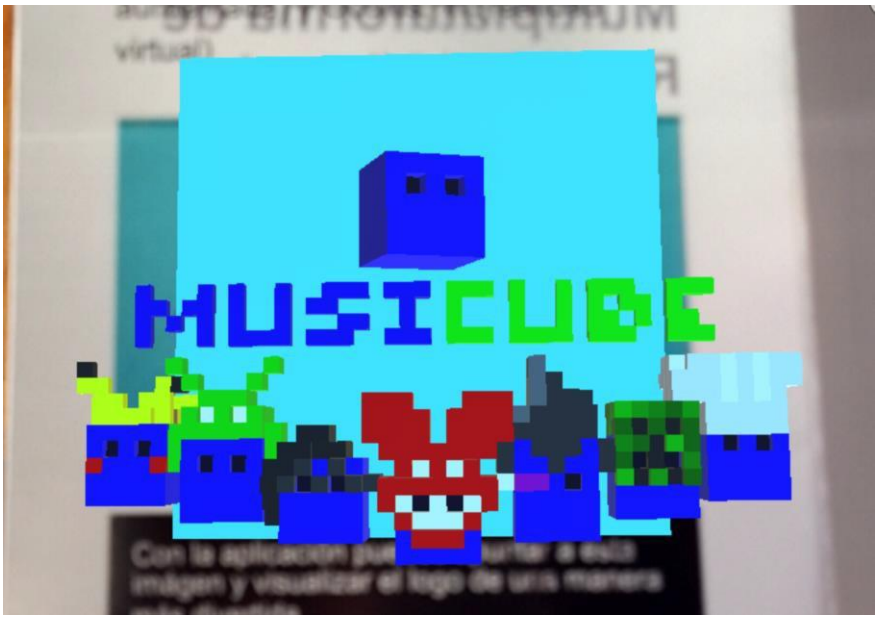

Figura 39. Interfaz móvil

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 27. *Administración de controlador general de la simulación*

Tarea	OE23 Administración de controlador general de la simulación	
Roles	Equipo de desarrollo	**Prioridad: 420**
Entrada	OG07 Controlador de realidad virtual	
Definición	Este controlador tiene la funcionalidad de administrar los demás scripts y controlar la jugabiilidad, este se encuentra vinculado al dispositivo de realidad virtual para que se ejecute las interacciones con el usuario.	

Ejecución Característica

En un objeto vacío se ubica el script que controla todos los scripts, manejando la jugabilidad y el momento de ejecución de cada uno, este objeto ira acoplado al modelo de realidad virtual.

Verificación

El script controla la jugabilidad correctamente.

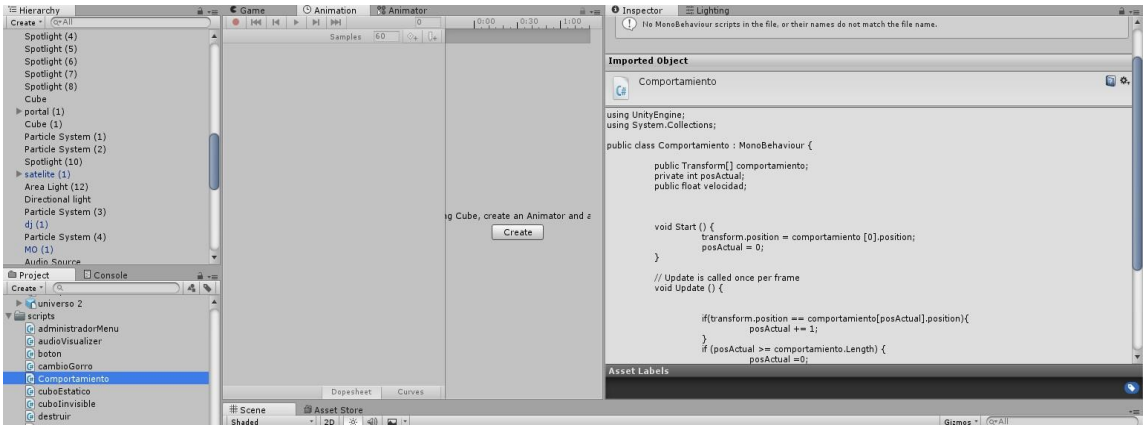

Figura 40. Control de jugabilidad

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 28. *Administración de controlador de integraciones*

Tarea	OE24 Administración de controlador de integraciones	
Roles	Equipo de desarrollo	**Prioridad: 430**
Entrada	OG07 Controlador de realidad virtual	
Definición	Se desarrolla controlador que permite unir todos los periféricos necesarios como los dispositivos de realidad virtual con el reconocimiento de las manos, y la entrada de AUX, para que no causen conflictos.	

Ejecución Característica

Este script se encarga de instancias todos los periféricos que se van a utilizar, en este caso se instanciaran 3, el Oculus Rift, el Leap Motion, y

por último la entrada AUX, para que podamos utilizarlos en la simulación (Okita, 2015, p. 102).

Verificación

Se puede observar funcionando las instancias.

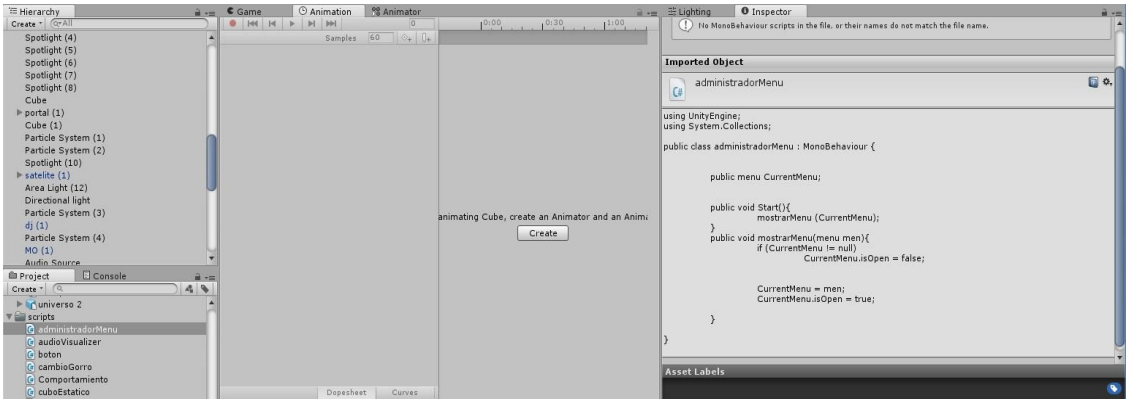

Figura 41. Controlador de integraciones

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 29. *Administración de controlador de validación*

Tarea	OE25 Administración de controlador de validación	
Roles	Equipo de desarrollo	**Prioridad: 440**
Entrada	OG07 Controlador de realidad virtual	
Definición	Se desarrolla un controlador que permita validar estados de los objetos activados, desactivados, en ejecución, y cancelados, esta validación se aplica con los modelos dinámicos y los estáticos dependiendo a la interacción que tengan con el usuario o con las frecuencias.	

Ejecución Característica

Este script se encarga de validar las funciones OnColissionEnter(), OnCollisionExit(),OnTriggerEnter();y OnTriggerExit(), valida en caso de cada función que comportamiento debe tener el modelo, en este caso se valida que dependiendo si lo que toca es una moneda, el puntaje aumenta.

Verificación

Se puede observar funcionando la validación con el trigger.

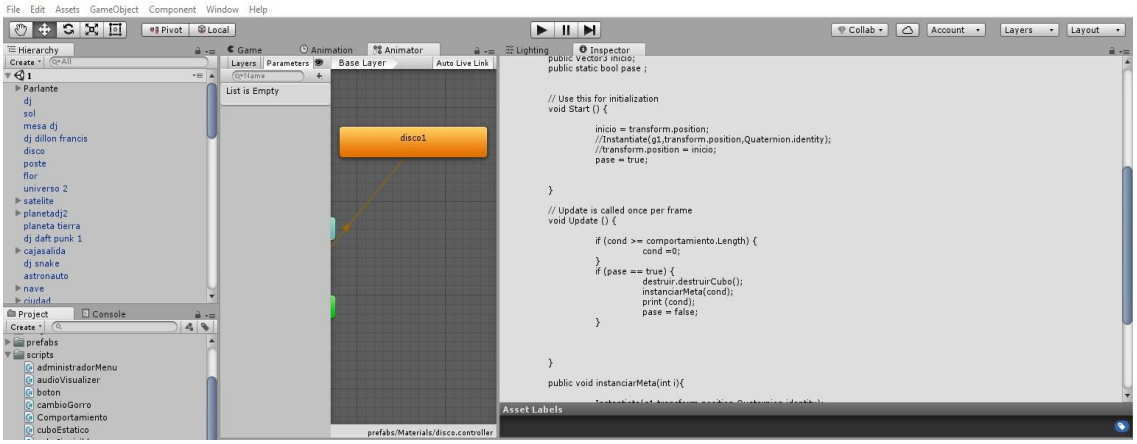

Figura 42. Script control de validación

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 30. *Exportación de modelos*

Tarea	OE26 Exportación de modelos	
Roles	Equipo de desarrollo	**Prioridad: 450**
Entrada	OG08 Migración con realidad aumentada	
Definición	Se deben exportar los modelos dinámicos, estáticos, y de entorno que se puedan utilizar ya que no se tiene tanta capacidad de renderizado, por lo que se exporta solo lo necesario.	

Ejecución Característica

Para poder realizar la migración con el celular, necesitamos exportar los modelos necesarios, no todos ya que en realidad aumentada no podemos apreciar todo el entorno, pero si los modelos que se encuentran dentro, por lo tanto, se exporto estos modelos para utilizarlos en la migración.

Verificación

Modelos exportados para migración móvil.

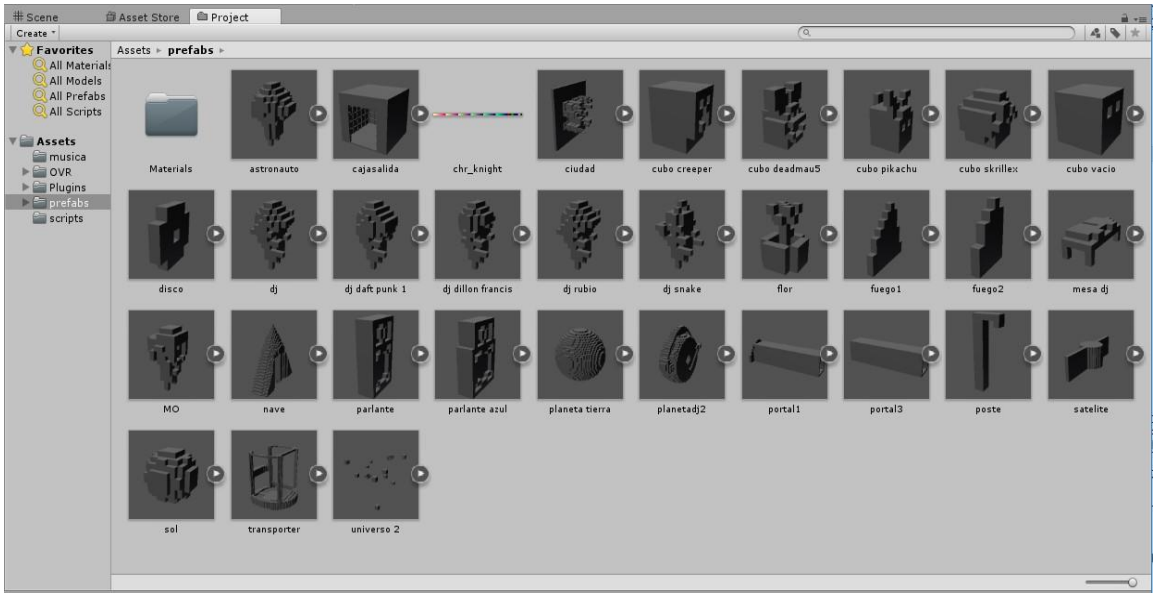

Figura 43. Modelos exportados

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 31.

Creación de Patrón de reconocimiento y base de datos

Tarea	OE27 Creación de Patrón de reconocimiento y base de datos	
Roles	Equipo de desarrollo	**Prioridad: 460**
Entrada	OG08 Migración con realidad aumentada	
Definición	Utilizando Vuforia creamos un patrón que sea amigable y referente a la simulación, y realizar una base de datos con los patrones que vamos a utilizar, esta base de datos se la exporta como asset para poderla importar desde Unity.	

Ejecución Característica

En la plataforma de Vuforia iniciamos una key, creamos una base de datos e ingresamos un patrón, este patrón tiene que ser completamente aumentable para una mejor experiencia (Morales, 2015, p. 53).

Verificación

Base de datos creada correctamente.

Figura 44. Base de datos Vuforia creada

Manejo de Problemas

Si el patrón no es suficientemente aumentable la aplicación tendrá problemas en obtener el patrón de la imagen por lo tanto no se mostrará en realidad aumentada, para arreglar esto se debe realizar un patrón más complejo del cual se puedan obtener más puntos de referencia**.**

Tabla 32. *Migración a realidad aumentada con Unity*

Tarea	OE28 Migración a realidad aumentada con Unity	
Roles	Equipo de desarrollo	**Prioridad: 470**
Entrada	OG08 Migración con realidad aumentada	
Definición	Unimos toda la base de datos, los modelos, y los scripts para que funcionen en un nuevo proyecto, realizamos un cambio de plataforma para celulares móviles, y generamos el Android Application Package (apk) que funcionara con el dispositivo.	

Ejecución Característica

Una vez obtenido el patrón ingresado, se descargó la base de datos exportándolo en formato package de Unity, y por último se importó en Unity con su respectiva key, y se une con los modelos importados del otro proyecto para su funcionamiento.

Verificación

Modelos exportados para migración móvil.

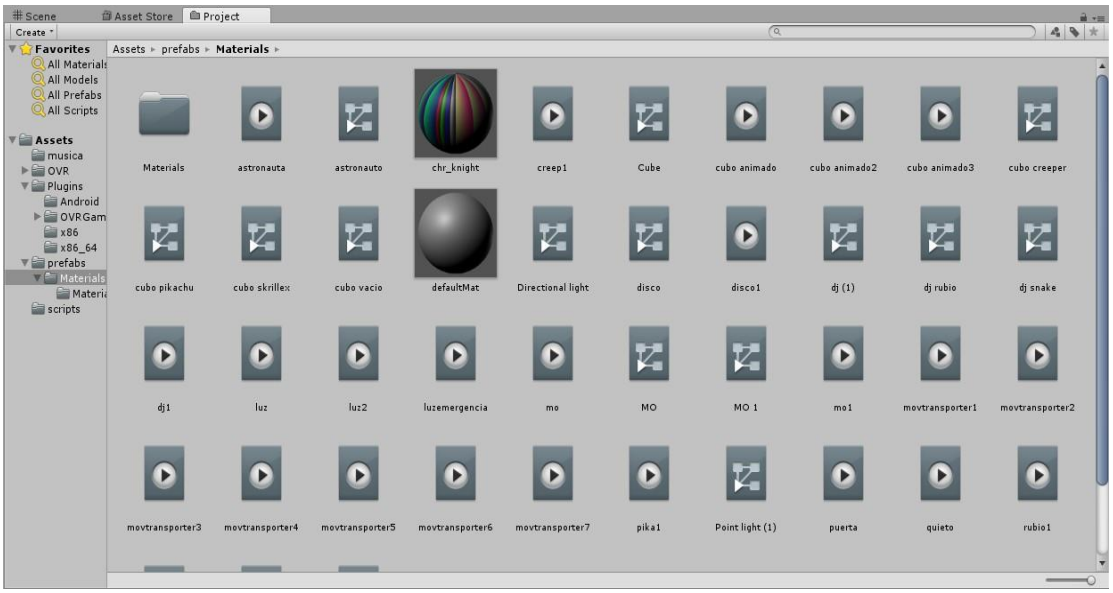

Figura 45. Exportación móvil

Manejo de Problemas

No hubo ningún Problema en esta ejecución**.**

Tabla 33.

Pruebas finales en aplicativo de realidad virtual

Tarea	OE29 Pruebas finales en aplicativo de realidad virtual	
Roles	Equipo de desarrollo, usuarios	**Prioridad: 500**
Entrada	OG09 Entregable final	
Definición	Se realizan pruebas para ver que toda la simulación se encuentre en completa funcionalidad.	

Ejecución Característica

Se realizan pruebas finales del funcionamiento de la simulación utilizando el Oculus Rift y el Leap Motion, utilizando un celular como entrada AUX.

Verificación

La simulación funciona correctamente.

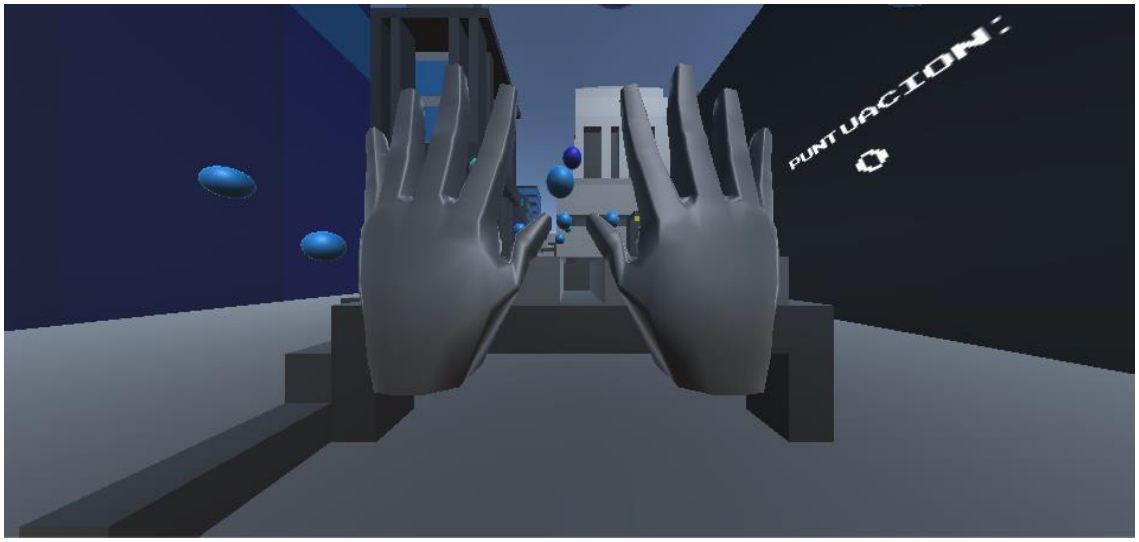

Figura 46. Simulación funcionando

Manejo de Problemas

Hubo pequeños cambios que se realizaron con éxito**.**

Tabla 34. *Pruebas finales en aplicativo de realidad aumentada*

Tarea	OE30 Pruebas finales en aplicativo de realidad aumentada	
Roles	Equipo de desarrollo, usuarios	**Prioridad: 510**
Entrada	OG09 Entregable final	
Definición	Se realizan las últimas pruebas de funcionalidad sobre la aplicación de dispositivos móviles con realidad	

Ejecución Característica

Se instaló el Android Application Package (apk) de prueba en un celular y una tablet, y se verifico el correcto funcionamiento de la aplicación en

realidad aumentada (Wiebe, 2015).

Verificación

La simulación funciona correctamente.

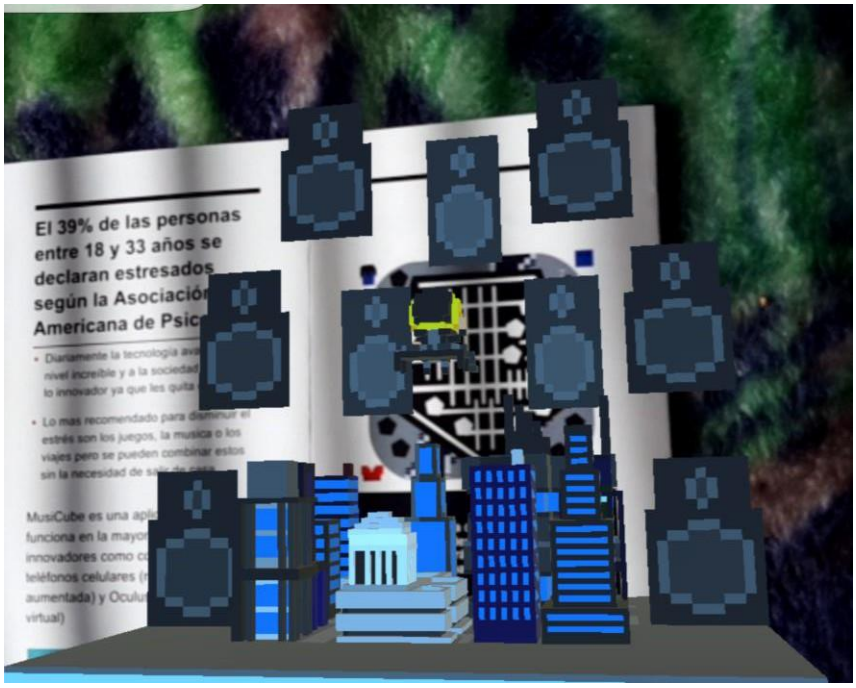

Figura 47. Simulación móvil funcionando

Manejo de Problemas

El apk no se había firmado de manera correcta por el bundle, esto se arregló especificando bien el bundle y firmando el apk**.**

Tabla 35. *Ejecutable final de escritorio*

Tarea	OE31 Ejecutable final de escritorio	
Roles	Equipo de desarrollo	**Prioridad: 520**
Entrada	OG09 Entregable final	
Definición	Tras haber realizados todas las pruebas con el usuario, se genera un ejecutable que será subido a una plataforma de distribución de manera gratuita.	

Ejecución Característica

Se generó un instalador que automatiza la implementación de todas los componentes y carpetas necesarias, y se probó el funcionamiento en un computador.

Verificación

La simulación funciona correctamente.

Manejo de Problemas

La simulación no funciono la primera vez debido a que faltaron archivos de insertar en él instalador**.**

Tabla 36. *Apk de Android*

Tarea	OE32 Apk de Android	
Roles	Equipo de desarrollo	**Prioridad: 530**
Entrada	OG09 Entregable final	
Definición	Luego de todas las pruebas con el dispositivo móvil, se procede a generar un apk que será subido a una plataforma de distribución con ningún costo.	

Ejecución Característica

Se generó un apk final ya para producción con su respectivo icono y bundle y se lo probo instalando y ejecutando en un celular.

Verificación

El apk funciona completamente.

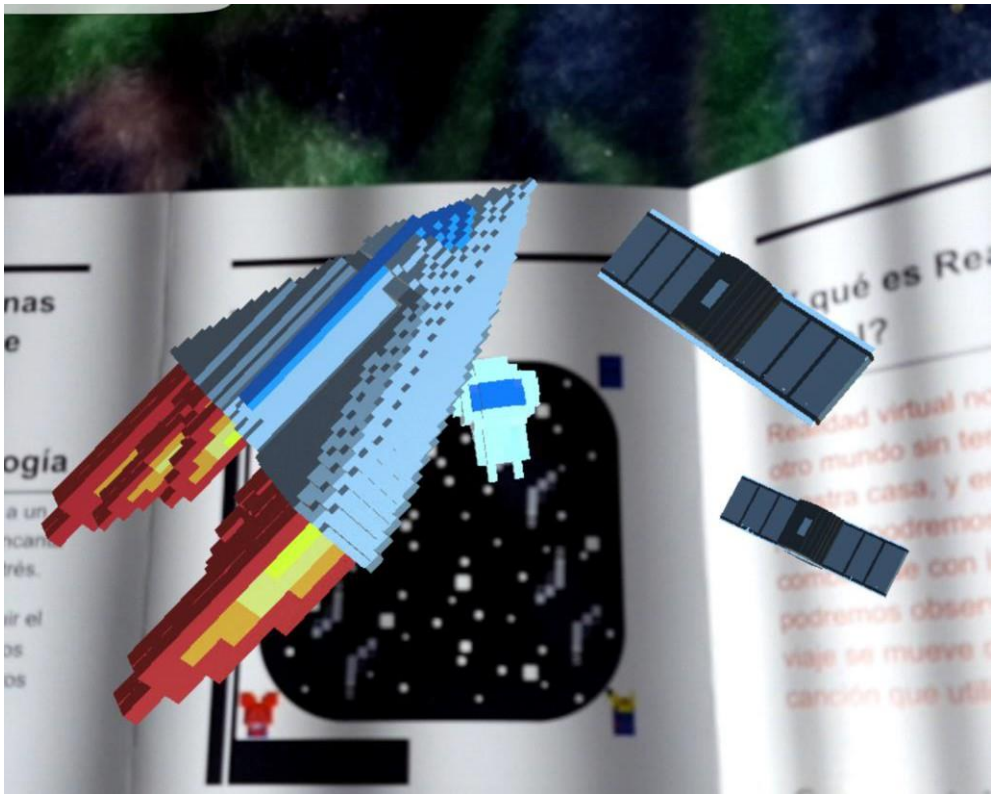

Figura 48. Apk MusiCube

Manejo de Problemas

El icono se ingresó primeramente en jpg lo que causó problemas visuales en el apk, esto se arregló ingresando un icono en formato png.

3.3.2 Estado actual del proyecto

El proyecto se encuentra finalizado en su elaboración y está listo para pasar a ser testeado antes de publicarlo

3.3.3 Medidas registradas

Cada iteración tuvo una duración aproximada de 2 días trabajando 4 horas por día.

3.3.4 Elementos positivos y negativos

Se registran muy pocos aspectos negativos durante las iteraciones como el versionamiento, pero en aspectos positivos incluye mucha investigación y

nuevo conocimiento.

3.3.5 Cumplimiento de los objetivos

Todas las iteraciones fueron realizadas correctamente y a tiempo tomando en cuenta el manejo de problemas ninguna iteración paso de los 3 días.

3.3.6 Mejoras al proceso

Las propuestas se realizarán en la parte beta ya que no hubo ningún cambio en las iteraciones ni en el plan de proyecto, por lo que estas tareas se la realizaran al reportar los resultados.

3.3.7 Determinar y Realizar cambios

Estos cambios serán determinados en las pruebas Beta y se realizarán como una tarea nueva.

3.4 Beta

3.4.1 Aspectos funcionales

En el caso de esta versión beta se lanzará una sola que contenga toda la funcionalidad del juego para solamente corregir los errores menores mas no la funcionalidad o el concepto.

Esta versión beta incluirá todos los aspectos funcionales indicados en la etapa de elaboración.

3.4.2 Medio de distribución

Para la distribución de la beta en Android se publicará en Google play una versión gratuita para corregir y comprobar la funcionalidad, en el caso de Oculus Rift este obtendrá un ejecutable que será distribuido físicamente a un reducido número de personas que probaran su funcionalidad.

3.4.3 Verificadores Beta

Para la versión de Android estará destinada para cualquier individuo con celular de gama medio-alta, para la versión de Oculus Rift estará destinada un grupo seleccionado de la Unidad de Innovación Tecnológica de la Universidad de las Américas.

3.4.4 Estado actual del proyecto

Tras haber realizado las pruebas en Android se comprobó que el estado del proyecto en la versión portable se encuentra completamente funcional, en la versión de Oculus Rift se presentó pequeños problemas de calibración y de ubicación en el plano 3D.

3.4.5 Reportar resultados

Tabla 37. *Reajuste Oculus Rift y ubicar plano 3D*

Tarea	OE33 Reajustar Oculus Rift y ubicar plano 3D	
Roles	Equipo de desarrollo	**Prioridad: 550**
Entrada	OG09 Entregable final	
Definición	Luego de todas las pruebas con el dispositivo Oculus Rift, se desarrolla el cambio previsto para que su funcionalidad sea completa, se lo distribuye como una actualización.	

Ejecución Característica

Se realizaron los cambios desde el editor de Unity, tras haber realizado los cambios se realiza un entregable como actualización a la versión final.

Verificación

La nueva versión funcionando correctamente.

Manejo de Problemas

No hubo ningún problema al realizar la actualización.

3.4.6 Evaluar y priorizar cambios

Luego de que se realizó la tarea se lo ubico con su respectiva prioridad y se realizó la actualización, y el entregable final puede salir en producción en la próxima etapa.

3.4.7 Pruebas de Usuario

Tabla 38. *Pruebas de Usuario*

Pruebas de Usuario	
Título	**PU01 Iniciar aplicación de escritorio.**
Proceso	Se inicia el ejecutable para 64 bits que abrirá la pantalla de resolución y configuración.
Estado	Se ejecutó exitosamente.
Título	**PU02 Cambiar la configuración.**
Proceso	En la pantalla del ejecutable se cambia la configuración de resolución.
Estado	Se realizó exitosamente.
Título	**PU03 Iniciar la simulación.**
Proceso	En la pantalla del ejecutable se presiona el botón para iniciar la simulación.
Estado	Se realizó exitosamente.
Título	**PU04 Iniciar interfaz Oculus.**
Proceso	Al momento de iniciar la simulación se inicia automáticamente la interfaz de realidad virtual de Oculus.

Estado	Se realizó exitosamente.
Título	**PU05 Manipular Menú.**
Proceso	Se puede manipular en la interfaz de realidad virtual con cada opción.
Estado	Se realizó exitosamente.
Título	**PU06 Empezar simulación**.
Proceso	En el menú se presiona iniciar para comenzar con la simulación.
Estado	Se realizó exitosamente.
Título	**PU07 Conectar cable aux.**
Proceso	Se conecta el celular mediante el cable aux para poder escuchar en la simulación la canción.
Estado	Se realizó exitosamente.
Título	**PU08 Jugar la simulación.**
Proceso	Con todo preparado se inicia la simulación y se observa el
Estado	Se realizó exitosamente.
Título	**PU09 Conseguir puntos.**
Proceso	Utilizando la mano se topan las monedas flotantes para obtener puntos.
Estado	Se realizó exitosamente.
Título	**PU10 Interacción con el entorno.**
Proceso	Se utiliza la mano para topar objetos en el entorno que se comportaran diferente al momento de toparlos.
Estado	Se realizó exitosamente.
Título	**PU11 Terminar la simulación.**
Proceso	Luego de haber terminado con la simulación automáticamente va al menú principal.
Estado	Se realizó exitosamente.
Título	**PU12 Instalar apk.**
Proceso	Se instala el apk en el celular para poder utilizar la aplicación
Estado	Se realizó exitosamente.
Título	**PU13 Iniciar la aplicación.**
Proceso	Se presiona en el acceso directo de la pantalla del celular para iniciar la aplicación.
Estado	Se realizó exitosamente.
Título	**PU14 manejo de interfaz móvil.**
Proceso	Se debe observar el menú e interactuar con él.
Estado	Se realizó exitosamente.
Título	**PU15 utilizar realidad aumentada en el móvil.**
Proceso	Se apunta la cámara hacia el patrón de reconocimiento para observar la realidad aumentada.
Estado	Se realizó exitosamente.

3.4.8 Pruebas de estrés

Las pruebas de estrés se realizaron probando en un computador con las siguientes especificaciones:

Procesador: AMD FX-8350

Memoria RAM: DDR3 16Gb

Tarjeta Gráfica: Nvidia gtx 1080

Puerto de salida: salida de video HDMI 1.3

Puertos de entrada: 4 puertos USB 3.0

Sistema Operativo: Windows 10 de 64 bit

Las pruebas obtenidas son las siguientes:

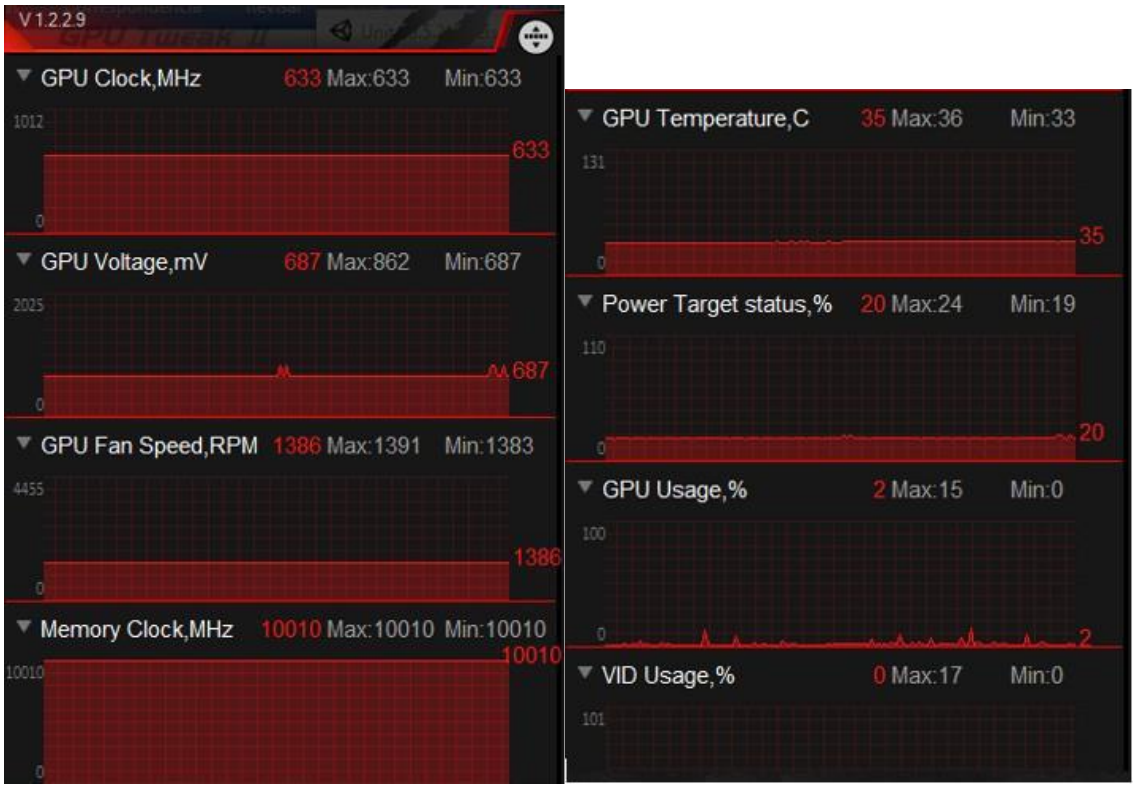

Figura 49. Pruebas de estrés

Con estas pruebas podemos observar que el consumo del procesador es muy grande en comparación con el consumo de la tarjeta gráfica, esto se debe a que el procesador no aguanta completamente la potencia de la

tarjeta gráfica, pero sigue teniendo un buen rendimiento en la simulación, tomando en cuenta eso se puede sacar requerimientos mínimos para el funcionamiento de la simulación los cuales son:

Procesador: AMD FX-8300 o Intel i5-4590

Memoria RAM: DDR3 8Gb

Tarjeta Gráfica: Nvidia gtx 970

Puerto de salida: salida de video HDMI 1.3

Puertos de entrada: 3 puertos USB 3.0 y 1 puerto USB 2.0

Sistema Operativo: Windows 7 de 64 bit en adelante

3.5 Gestión de Riesgos

Tabla 39.*Versionamiento plataforma*

R01 Versionamiento plataforma	
Identificar riesgos	Se identifican los posibles riesgos de cambio de versión de Unity, que pueden dar problemas de compatibilidad.
Evaluar riesgos	Se evalúa que el riesgo puede ocurrir, pero no es un riesgo de prioridad alta.
Estrategias para mitigar los riesgos	Para mitigar este riesgo se optó por tener una actualización estable para el desarrollo sin tener la necesidad
Planes de contingencia	No se realiza un plan de contingencia ya que el riesgo es menor.
Monitorear riesgos	Se monitorea las versiones de las herramientas para el desarrollo y evitar tener futuros problemas de compatibilidad.

Tabla 40. *Calibrar periféricos*

R02 Calibración periféricos	
Identificar riesgos	Se identifican los posibles riesgos al momento que se calibre un dispositivo.
Evaluar riesgos	Se evalúa que el riesgo puede ocurrir, pero no es un riesgo de prioridad alta.
Estrategias para mitigar los riesgos	Para mitigar este riesgo se optó por guardar una calibración de cada dispositivo para no tener que calibrar múltiples veces.
Planes de contingencia	No se realiza un plan de contingencia ya que el riesgo es menor.
Monitorear riesgos	Se monitorea las calibraciones de los periféricos para revisar que no se ha cambiado de calibración en ningún dispositivo.

Tabla 41. *Control de errores*

R03 Control de errores	
Identificar riesgos	Se identifican los posibles riesgos de los errores de códigos.
Evaluar riesgos	Se evalúa que el riesgo puede ocurrir, y su prioridad es media.
Estrategias para mitigar los riesgos	Para mitigar este riesgo se optó por utilizar la consola del Visual Studio y no la de Unity, para poder controlar errores al momento de escribir y no al de correr el editor.
Planes de contingencia	No se realiza un plan de contingencia ya que el riesgo es menor.
Monitorear riesgos	Se monitorea mediante la consola para poder corregir los errores lo más rápido posible.

Tabla 42. *Diseño*

R04 Diseño	
Identificar riesgos	Se identifican los posibles riesgos de los errores en el diseño.
Evaluar riesgos	Se evalúa que el riesgo puede ocurrir, y su prioridad es media.
Estrategias para mitigar los riesgos	Para mitigar este riesgo se optó por utilizar un control de avances comparando con el diseño actual utilizado, de esta manera identificar faltantes o riesgos en el diseño.
Planes de contingencia	En caso de encontrar un riesgo, se procede inmediatamente a realizar el cambio necesario en el diseño original para cumplir con la necesidad.
Monitorear riesgos	Se monitorea mediante las pruebas de usuario si es necesario algún cambio constantemente.

Tabla 43. *Jugabilidad*

R05 Jugabilidad	
Identificar riesgos	Se identifican los posibles riesgos de compatibilidad de controles sin tener interferencias.
Evaluar riesgos	Se evalúa que el riesgo puede ocurrir, y su prioridad es media.
Estrategias para mitigar los riesgos	Para mitigar este riesgo se optó por verificar cada vez que se realiza un cambio de jugabilidad que se controle que no cause conflicto con el resto de jugabilidad.
Planes de contingencia	En caso de encontrar un riesgo, se procede inmediatamente a realizar el cambio necesario para hacer compatible las funcionalidades de la jugabilidad.
Monitorear riesgos	Se monitorea mediante pruebas de funcionamiento después de cada tarea realizada.

3.6 Cierre

3.6.1 Entregable Definido

El entregable será distribuido mediante las plataformas Google Play para Android y en App Store para IOS, será distribuido de manera gratuita y funcionará en versiones de Android 5.0+ e IOS 7+ con el nombre de MusiCube.

3.6.2 Entregable Realizado

El entregable realizado esta generado en un formato de apk para la versión de Android y en formato ipa para la versión de IOS mediante el compilador de Unity para producción.

3.6.3 Entregable Validado

El entregable es aprobado en la versión de Android por lo que el entregable se encuentra en producción y funcionando.

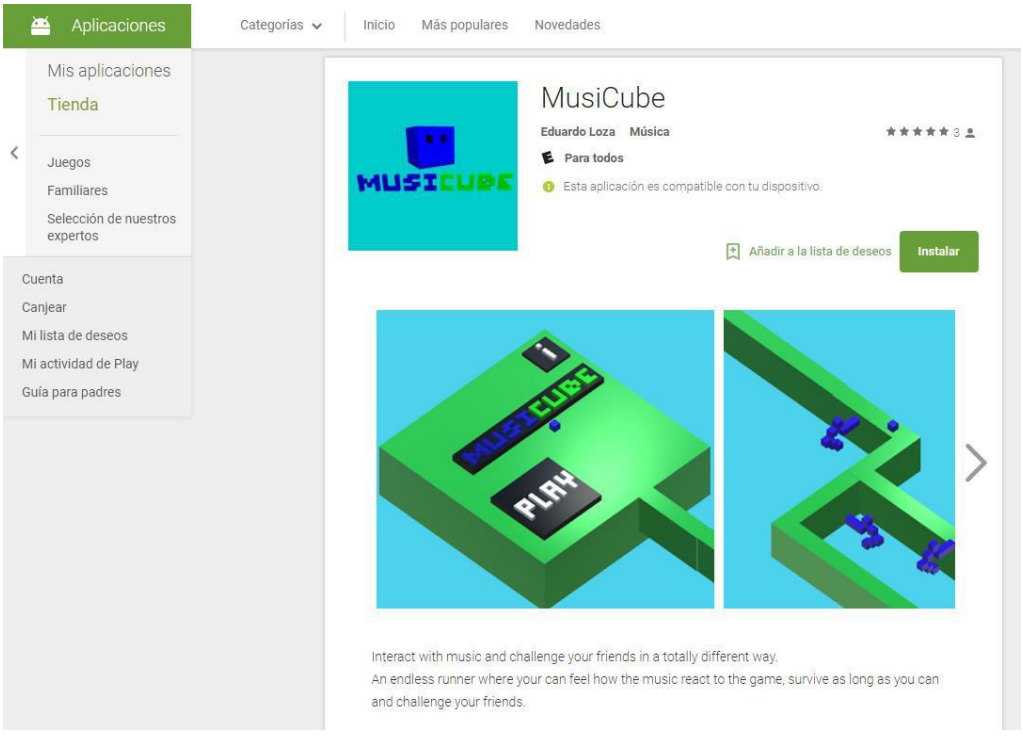

Figura 50. Entregable Musicube en Google Play

3.6.4 Registro lecciones aprendidas

- Con el proyecto se pudo aprender la importancia de seguir una metodología paso por paso para el desarrollo de la simulación.
- Se pudo aprender la importancia de realizar una versión beta para poder controlar errores y monitorear problemas.
- Se aprendió a utilizar la gestión de riesgos para poder monitorear errores que puedan suceder en cualquier momento del desarrollo.
- Se pudo aprender el proceso para desarrollar tareas específicas que llevaran cierto tiempo de desarrollo.

3.6.5 Mejoras propuestas a la metodología

- Se propone mejorar la parte de pruebas de usuario utilizando dos enfoques, el de la prueba de un programador y la de un usuario final.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

El desarrollo del prototipo involucro muchas horas de investigación para lograr desarrollar en herramientas totalmente nuevas, la innovación y la investigación van de la mano en cualquier campo, como el educacional o el profesional, lo cual es muy importante mantenerse actualizado en los conocimientos para hacer posibles sistemas nuevos a cada momento.

Se tomó en cuenta que para lograr la integración entre el sensor de Leap Motion y Oculus Rift se tuvo que modificar el sdk de ambos, para permitir visualizar los patrones de las manos en realidad virtual.

Es importante que para reconocer las frecuencias de una canción se utilizó la entrada auxiliar para poder recibir esta información, y clasificarla en un formato numérico, y a su vez por calidad de sonido también se lo reconoce localmente en el proyecto, luego se utilizó los fotogramas con este formato para poder manipularlos acorde con las frecuencias y así producir objetos que cambian sus propiedades físicas con la música.

Se tomó en cuenta que el entorno de la simulación funcionaria dependiendo de los valores más grandes o más pequeños de las frecuencias para llegar a obtener un entorno inteligente que reaccione e interactúe dependiendo de la canción por lo que el reconocimiento a tiempo real tuvo que ser renderizado en los fps al mismo tiempo que se anima un modelo dinámico.

4.2 Recomendaciones

Las integraciones entre diferentes sensores muchas veces no son compatibles, y es recomendable entender y analizar el sdk para que al

momento de realizar una integración no de problemas de compatibilidad o mal funcionamiento debido al choque de funciones.

El código de reconocimiento de frecuencias es escalable y se podría utilizar no solo para que el entorno interactúe con una canción, sino que también se puede utilizar para reconocimiento de voz, de palabras e interacciones con cualquier objeto o script que utilice valores numéricos.

Un entorno inteligente que reacciona a la música como este se podría utilizar e incrementar para basarse en el estado o género de un videojuego o para realizar un reproductor musical más dinámico.

REFERENCIAS

Creative. (2017). *Digital Tutors*. Recuperado el 12 de enero de 2017 de http://www.digitaltutors.com/software/Unity-tutorials

Ephtracy. (2017).*Magica Voxel*. Recuperado el 4 de enero del 2017 de https://ephtracy.github.io/

Leap Motion. (2017).*Leap Motion Developer*. Recuperado el 5 de enero del 2017 de https://developer.leapmotion.com/

Maxon. (2017).*Cinema 4D*. Recuperado el 2 de enero del 2017 de https://www.maxon.net/es/productos/cinema-4d/cinema-4d/

Microsoft. (2017).*Visual Studio*. Recuperado el 2 de enero del 2017 de https://www.visualstudio.com/es/?rr=https%3A%2F%2Fwww.google.com%2F

Morales, C. (2015). *Desarrollando aplicaciones de realidad aumentada con Unity 3D*. Recuperado el 10 de febrero de 2017 de https://books.google.com.ec/books?id=kUVKCgAAQBAJ&printsec

Newzoo. (2016). *2016 Global Games Market Report*. Recuperado el 7 de febrero de 2017 de https://newzoo.com/insights/articles/global-games-market-reaches-99-6-billion-2016-mobile-generating-37/

Oculus. (2017).*Oculus Rift*. Recuperado el 2 de enero del 2017 de https://www.oculus.com/

Okita, A. (2015). *Aprendiendo programación C# con Unity 3D*. Recuperado el 12 de marzo de 2017 de https://books.google.com.ec/books?id=4wIZCwAAQBAJ&pg=PA6&dq

SUM. (2017).*Sum para desarrollo de videojuegos*. Recuperado el 9 de enero de 2017 de http://www.gemserk.com/sum/

Unity. (2017).*Unity 3D*. Recuperado el 12 de enero del 2017 de https://Unity3d.com/es/learn/tutorials

Vuforia. (2017).*Augmented Reality*. Recuperado el 12 de enero del 2017 de https://www.vuforia.com/

Watkins, A. (2012). *Creating games with Unity and maya*. Burlington: Elsevier. Wiebe, R. (2015). *Unity iOS Essentials*. Boston: Packt.