



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO ECATEPEC

**Uso de la Realidad Virtual para el estudio del movimiento
unidimensional: una actividad didáctica**

TESIS:

PARA OBTENER EL TÍTULO DE MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

Jesús Eduardo Flores Lara

ASESOR ACADÉMICO:

M. en I.S.C. Alejandra Morales Ramírez

TUTORES ADJUNTOS:

Dr. en C. Rodolfo Zolá García Lozano

M. en C. C. Juan de Jesus Amador Reyes

Página en blanco

ÍNDICE

INTRODUCCIÓN	8
CAPÍTULO 1 MARCO TEÓRICO.....	14
1.1 Realidad Virtual	14
1.2 Clasificación de la RV.....	15
1.2.1 RV Inmersiva.....	15
1.2.2 RV no inmersiva o de escritorio	15
1.3 Características de la RV	16
1.3.1 Inmersión	16
1.3.2 Interacción.....	17
1.3.3 Imaginación	17
1.4.1 CAVE	17
1.4.2 Guantes de RV	18
1.4.3 Lentes de Realidad Virtual.....	18
1.4.4 Caminadora de Realidad Virtual.....	19
1.5 Uso de la RV en ambientes educativos	20
1.6 La RV y las teorías del aprendizaje	21
1.7 Herramientas de desarrollo	22
1.7.1 Herramientas de texturizado	22
1.7.1.1 Adobe Photoshop.....	22
1.7.2 Herramientas de modelado	23
1.7.2.1 Autodesk Maya.....	23
1.7.3 Herramientas de animación	24
1.7.3.1 Daz 3D	24
1.7.3.2 Iclone	25
1.7.3.3 Iclone3DXchange.....	26
1.7.4 Adobe Audition	27
1.7.5 Motores gráficos	28
CAPÍTULO 2 METODOLOGÍA DE DESARROLLO.....	31
2.1 Fase de Análisis	32
2.1.1 Identificación de requerimientos.....	32

2.1.2	Propuesta de desarrollo	32
2.1.3	Especificación de requerimientos	32
2.2	Fases de Diseño	33
2.2.1	Modelado conceptual	33
2.2.2	Modelado de Navegación	33
2.2.3	Descripción de componentes.....	33
2.2.4	Modelado de Eventos	34
2.3	Fase de Desarrollo	34
2.3.1	Producción de recursos artísticos	35
2.3.2	Animación e Integración	35
2.3.3	Producción de recursos tecnológicos.....	36
2.3.4	Pruebas.....	36
2.4	Fase de Evaluación	36
2.5	Fase de Cierre y Entrega Final	36
CAPÍTULO 3 IMPLEMENTACIÓN DE LA METODOLOGÍA.....		38
3.1	Fase de Análisis	38
3.1.1	Identificación de requerimientos.....	38
3.1.2	Propuesta de desarrollo	39
3.1.3	Especificación de requerimientos	41
3.2	Fase de Diseño	42
3.2.1	Modelado conceptual	42
3.2.2	Modelado de navegación.....	49
3.2.3	Story Board.....	50
3.2.4	Descripción de componentes.....	53
3.2.5	Modelado de eventos	54
3.3	Fase de Desarrollo.....	56
3.3.1	Producción de recursos artísticos	56
3.3.1.1	Modelado	56
3.3.1.2	Optimización	59
3.3.2	Producción de audio.....	62
3.3.3	Creación y optimización de personajes de Daz3D	64
3.3.4	Animación.....	69
3.3.5	Skybox	74

3.3.6	Integración	80
3.3.7	Integración de lep motion.....	87
3.3.8	Programación	94
3.3.5.1	Identificación de manos Leap motion.....	94
3.3.5.2	Reciclaje de componentes	95
3.3.5.3	Programación de gravedad	97
3.3.5.4	Programación de cámaras.....	97
3.3.6.5	Programación del tutorial de ayuda para el uso del Leap Motion.....	99
3.3.5.6	Programación de máquinas de estado y animación	101
3.3.9	Pruebas de integración.....	104
3.3.10	Empaquetado	107
3.4	Fase de Evaluación y cierre	110
3.4.1	Valoración por el cliente / usuario.....	110
CAPÍTULO 4 CONCLUSIONES		111

ÍNDICE DE FIGURAS

Figura 1.RV inmersiva.....	15
Figura 2.RV no inmersiva.....	16
Figura 3.Triangulo de la RV.	16
Figura 4.Uso del C.A.V.E.	18
Figura 5.Guantes de RV.	18
Figura 6.Htc lentes de RV.	19
Figura 7.Caminadora de realidad virtual.....	19
Figura 8.Fotoshop.....	23
Figura 9.Interface Autodesk Maya	24
Figura 10.Interface Daz 3D.....	25
Figura 11.Interface Iclone	26
Figura 12.Interface Iclone 3DXange	27
Figura 13.Interface Adone Audition	28
Figura 14.Metodología de desarrollo evolutivo para escenarios tridimensionales (Morales, et. al, 2015)	31
Figura 15.Fase de desarrollo	35
Figura 16.Mapa distribución general para el escenario de la plasa de pisa	43
Figura 17.Bosquejo general de distribución de la plaza de pisa	43
Figura 18.Mapa distribución general de los escenarios de experimentación (Torre de pisa, Jupiter, Marte, Luna y Cámara de vacío)	44
Figura 19.Diseño de la vista del mirador de la torre de pisa desde el planta tierra	44
Figura 20.Diseño de vista del mirador de la torre desde la Luna.....	45
Figura 21.Diseño de vista del mirador de la torre desde Marte	45
Figura 22.Diseño de vista del mirador de la torre desde Júpiter	46
Figura 23.Diseño de vista de la Cámara de vacío de la NASA	46
Figura 24.Planos de torres de pisa	47
Figura 25.Planos de catedral de pisa.....	47
Figura 26.Torre de pisa desde Streetview.....	48
Figura 27.Diagrama de navegación de la aplicación	49
Figura 28.Diagrama de navegación de subrutina de experimentación	49
Figura 29.Perspectiva en primera persona leap.	50
Figura 30.Base torre	57
Figura 31.Cortes torre	57
Figura 32.Pisos de la torre.....	58
Figura 33.Arcos de la torre.....	58
Figura 34.Arcos correctos de Pisa	59
Figura 35.Agregado de ejes.....	59
Figura 36.Uv map torre de pisa.....	60
Figura 37.Agregado de imagen	60
Figura 38.Corte de imágenes	61
Figura 39.Modelo texturizado.....	61

Figura 40. Dispositivos de grabación.....	62
Figura 41. Vista de ondas de sonido	62
Figura 42. Vista de espectro.....	63
Figura 43. Vista normalizada	64
Figura 44. Anatomía en Daz 3D	64
Figura 45. Selección en Daz 3D	65
Figura 46. Facekey en Daz 3D.....	65
Figura 47. Personaje en simplygon	66
Figura 48. División del personaje en simplygon.....	67
Figura 49. Vista comparativa de optimización	68
Figura 50. Vista comparativa de la maya	68
Figura 51. Integración de huesos	69
Figura 52. Inicio de animación	70
Figura 53. Conexión con kineckt	70
Figura 54. Sincronización kineckt y iclone.....	71
Figura 55. Secuencia de animación.....	71
Figura 56. Edición de animación corporal	72
Figura 57. Animación facial	72
Figura 58. Integración de elementos en fbx	73
Figura 59. Exportación de fbx.....	73
Figura 60. Skybox lunar	74
Figura 61. Armado de skybox en unity.....	75
Figura 62. Borrado de margen	76
Figura 63. Mapa de cubemap	76
Figura 64. Creación de textura	77
Figura 65. Toma de imagen de skybox.....	77
Figura 66. Ordenado de imágenes	78
Figura 67. Plantilla en photoshop	78
Figura 68. Cubemap finalizado.....	79
Figura 69. Cubemap en Unity.....	79
Figura 70. Cubemap en ejecución	80
Figura 71. Proyecto unity	81
Figura 72. Terreno	81
Figura 73. Estructura exportada a Unity	82
Figura 74. Escenarios en Unity	82
Figura 75. Personaje en Unity	83
Figura 76. Material cornea	83
Figura 77. Material Eye reflection.....	84
Figura 78. Material Pestañas.....	84
Figura 79. Paquete de texturas Daz	85
Figura 80. Transparencia en textura	86
Figura 81. Material Pestañas final.....	87
Figura 82. Carpetas de “Core Assets”	88
Figura 83. Detección de escritorio	89

Figura 84.Adaptador Oculus.....	89
Figura 85.Script “PluginLeapNotice”	90
Figura 86.Script “Zona de interacción”	90
Figura 87.Inspector de “PinchingHand”	91
Figura 88.Script Mouse Look.....	92
Figura 89.Script Mouse Look (cuerpo)	92
Figura 90.Script Mouse Look (cámara).....	93
Figura 91.Script Mouse Look (cámara).....	93
Figura 92.Script Mouse Look detección de mano	94
Figura 93.Script Mouse Look seguimiento de mano	95
Figura 94.Script LeapSuperHeroController detección de mano derecha	95
Figura 95.Script de destructor con condición	96
Figura 96.Script de instanciador con condición	96
Figura 97.Colisionadores de reciclaje.....	97
Figura 98.Script de gravedad.....	97
Figura 99.Script cambio de camara.....	98
Figura 100.Colisionadores de cambio de cámara	98
Figura 101.Texturas con render	99
Figura 102.Cámaras de render.....	99
Figura 103.Tutorial leap.	100
Figura 104.Script tutorial.....	100
Figura 105.Maquina de estado	101
Figura 106.Script de transición.....	102
Figura 107.Eventos dentro la animación.....	103
Figura 108.Script de audio.	103
Figura 109.Integración de audio	104
Figura 110.Ejecución de la aplicación	105
Figura 111.Opciones de calidad	107
Figura 112.Opciones de sistema operativo.....	108
Figura 113.Selección de icono.....	108
Figura 114.Selección de icono.....	109
Figura 115.Empaquetado final	109

ÍNDICE DE TABLAS

Tabla 1.Tabla de comparación de motores gráficos de representación.....	29
Tabla 2.Tabla de especificaciones.	32
Tabla 3.Descripción de componentes	34
Tabla 4.Extracto de contenido del programa de la materia ciencias del segundo año de secundaria	39
Tabla 5.Guion Bienvenida	40
Tabla 6.Guión torre	40
Tabla 7.Guión luna	40
Tabla 8.Guión marte.....	41
Tabla 9.Guión júpiter.....	41
Tabla 10.Guión cámara de vacío	41
Tabla 11.Requisitos funcionales y no funcionales.....	42
Tabla 12.Story board de escena introductoria.....	51
Tabla 13.Descripción de objetos	53
Tabla 14.Descripción de espacios arquitectónicos	53
Tabla 15. Identificación de secuencias de audio.....	54
Tabla 16.Secuencia de eventos de desplazamientos.....	54
Tabla 17.Secuencia de eventos para dispositivo Leap Motion durante la experimentación	54
Tabla 18. Secuencia de eventos con objetos	55
Tabla 19.Comparativa de optimización.....	67
Tabla 20.Tabla de errores.....	105

INTRODUCCIÓN

Hoy en día la tecnología se ha vuelto un factor de gran trascendencia en nuestra vida, el uso de las computadoras ha cambiado la forma en que trabajamos y llevábamos a cabo actividades. Por ejemplo, la evolución de los dispositivos móviles, ha transformado la manera en que nos comunicamos y entretenemos, así como este caso se pueden encontrar miles y en la educación esto no ha sido distinto.

Por ejemplo, los profesores actualmente no solo usan libros para apoyarse en sus clases, si no también utilizan grabaciones, videos, pizarrones inteligentes y revistas electrónicas, así como redes sociales (Twitter, Facebook, YouTube etc.) para seguir en contacto con sus alumnos e incluso utilizan Skype para hacer video llamadas y conferencias en línea (Knutzen & Kennedy, 2012).

Otra herramienta práctica pero complicada, que ha empezado a relucir en el ámbito educativo y atraído una atención especial por su potencial es la Realidad Virtual (RV) (Kastoudi, 2011), definida como un sistema informático generado por computadora (Zhan, 2011) que puede simular de manera completa o parcial un ambiente (situación) real o ficticio (Gurau, 2007). Y se clasifica según el nivel de inmersión con el usuario en: a) RV no inmersiva o de escritorio, en la que el usuario mediante una computadora de escritorio puede ver el espacio virtual y navegar en él a través del teclado y el ratón; y b) La RV inmersiva, que ofrece al usuario inmersión total mediante el uso de diferentes equipos de interacción como los sensores, guantes, lentes, diademas e incluso CAVEs (Zhan, 2011).

Además, la RV posee tres características fundamentales: inmersión, interacción e imaginación (Zhan, 2011; Shi, 2004). La inmersión es la característica más destacada de las tres, porque es donde el usuario encuentra una experiencia real, otorgándole la sensación de encontrarse dentro de los entornos tridimensionales. Dicha inmersión es determinada por las imágenes, la interacción, el comportamiento de los objetos y los entornos que dependiendo de la situación deberán ser los más semejante posible al mundo real.

La interacción por su parte implica la posibilidad de tener control sobre el sistema creado y, por ende, que los cambios producidos en el mundo virtual dependan del usuario y no de una programación previa.

Finalmente, la imaginación que implica no solo una demostración de contenido multimedia, sino una herramienta de diseño, que refleje el pensamiento del diseñador de una manera visual, por ejemplo, antes de la exhibición de un edificio, la primera cosa que se debe hacer es la estructura a detalle de la construcción a través de una gran cantidad de boceto (Zhan, 2011).

Características que han permitido que la RV ofrezca beneficios significativos en numerosas áreas del conocimiento como el entretenimiento, la formación, la difusión, la capacitación, el diseño, la terapia y la educación.

En cuanto a entornos virtuales desarrollados con RV enfocados a la educación se pueden distinguir los siguientes trabajos: “Aplicación de RV en la educación a distancia” (Zhan, 2011), “Aplicación de RV en la educación médica” (Shi, 2004), “Aplicación y perspectiva de la tecnología de realidad virtual en la educación ideológica universitaria” (Hui-Zhen & Zhong-Fa, 2013), “Diseño e implementación de un mundo virtual tridimensional multiusuario para el aprendizaje del lenguaje” (Ibañes, et al., 2010), “Diseño de RV inmersiva para la educación de geometría” (Kaufmann, & Schmalstieg, 2006), “Mundos virtuales en tercera dimensión y enseñanza a distancia: casos de estudios de mundos activos como medio para la educación a distancia” (Dickey, 2005), “Diseño de un entorno para el desarrollo de la competencia digital docente en estudiantes universitarios: usabilidad, adecuación y percepción de utilidad” (Antonietti, Rasi, Imperio and Sacco, 2000), “Un entorno de aprendizaje virtual 3d para la enseñanza de la química en la escuela secundaria” (Shudayfat, Moldoveanu & Moldoveanu, 2012) y “Realidad virtual como entorno de aprendizaje en el contexto EFL de Irán: personal, técnica y pedagógica” (Rahimi, Golshan & Mohebi, 2013).

Dichos trabajos de RV han demostrado ofrecer ventajas pedagógicas (Rahim, 2014; Antonietti, 2000) como las siguientes:

- Elevan el acceso a la capacitación.
- Ayudan a las personas a tener una visión global del tema de interés o que se encuentran estudiando.
- Ayudan a la comprensión de conceptos y temas.
- Motivan al usuario a seguir aprendiendo.
- Animar a la participación más activa del alumno en el aprendizaje del tema.
- Ayudan a los estudiantes a experimentar directamente algunas propiedades físicas de objetos y eventos y a darse cuenta de las implicaciones reales de tales propiedades.
- Permite que el principiante proceda con una experiencia según su propio camino.

Sin embargo, a pesar de que la RV ofrece ventajas pedagógicas y una plataforma totalmente nueva en el proceso de enseñanza – aprendizaje, en la cual se puede generar una enorme gama de actividades e ideas para implementar teorías y prácticas o una combinación de ellas (Kaufmann, & Schmalstieg, 2006), para facilitar la representación de fenómenos que no serían posibles experimentar en un aula o en un laboratorio. En México son pocos los trabajos que se han creado con RV en el área de educación.

En este sentido, se buscó un área curricular de oportunidad para desarrollar una aplicación didáctica, lo cual se llevó a cabo a través del análisis de los resultados de la prueba ENLACE 2012 que evalúa a los estudiantes desde tercer año de primaria hasta tercero año de secundaria para conocer el nivel educativo que hay en México en diversas áreas del conocimiento como son: matemáticas, español, ciencias e historia. Sin embargo, para delimitar el presente trabajo solo se tomaron en cuenta los resultados del área de ciencias a nivel secundaria.

De manera general se encontró lo siguiente:

- Cerca de un 61.7% de los estudiantes de nivel secundaria que realizaron esta prueba presentaron un nivel “elemental” en el área de ciencias ubicándolos por debajo del rango del dominio recomendado, ya que solo conocen lo básico, es decir requieren fortalecer la mayoría de los conocimientos y desarrollar aún más sus habilidades.
- De los 761,073 alumnos que hicieron la prueba en el Estado de México, se halló que 17.4% de los alumnos se encuentran en el rango de "insuficiente" en dicha área.
- Comparando los resultados obtenidos en el 2008 con los del 2012 se aprecia un solo un aumento del 0.3% al 0.7% en el nivel “excelente”, donde se encuentran aquellos estudiantes que conocen todos los conceptos y son capaces de integrarlos para resolver problema de su vida cotidiana. En cambio, el nivel "insuficiente" subió un 4%, 10 veces más que los alumnos en nivel “excelente”.
- En lo que respecta a los resultados por tema de conocimiento en área de ciencias menos de la mitad (40%) de los estudiantes se encuentra en un nivel “elemental”, en los temas: sociedad y tecnología; y manifestaciones de la materia.
- Mientras que el 60% de los alumnos presentan problemas en los tópicos: descripción del movimiento, leyes del movimiento, descripción de la materia, ubicándose en el rango de “insuficiente”, considerados como aquellos estudiantes que solo tienen la noción de los conceptos.

A pesar de que existe un aumento de alumnos en el grado de “excelente” a nivel nacional, es insignificante al contrastar con el aumento del grado de “insuficiente”, al igual haciendo una comparación más afondo de las dos aplicaciones (2008 y 2012) existe un aumento en el grado de “bueno”, pero no lo suficiente para equilibrar el grado de alumnos en situación “elemental” o “insuficiente”.

Por otro lado, los resultados de la prueba de PISA del 2015 que es aplicada por la Organización para la Cooperación y el Desarrollo Económico (OCDE) y que evalúa el dominio en matemáticas, lectura y ciencias a estudiantes de entre 15 y 16 años de edad, independientemente del grado que estén cursando.

Arrojo los siguientes resultados:

- México se posicionó por debajo del promedio mundial en el puesto número 56 de los 72 que toman la prueba, dejándolo entre los países de América Latina con peor rendimiento académico.
- En promedio los alumnos obtuvieron 416 puntos en el área de ciencias, dejando a México por debajo del promedio de la OCDE (493) y a un nivel similar al de Colombia, Costa Rica, Georgia, Montenegro, Qatar y Tailandia. Los jóvenes tienen una diferencia de más de 70 puntos por debajo de los estudiantes en Portugal y España, y una diferencia entre 20 y 60 puntos por debajo de los estudiantes en Chile y Uruguay, pero se sitúan por encima de los estudiantes Brasil, la República Dominicana y Perú.
- Un poco más del 20% de los alumnos, no alcanzaron el nivel mínimo de competencias en el área de ciencias. Esto quiere decir que los alumnos no son capaces de hacer uso básico del conocimiento de ciencias para identificar una respuesta apropiada, interpretar los datos e identificar las preguntas emergentes de un experimento simple.
- Menos del 1% de los alumnos, se encuentran en el grado de excelencia en el área de ciencias (sin tener ningún avance desde el 2006). En estos niveles, los estudiantes pueden aplicar los conocimientos y habilidades científicas, con autonomía y creatividad en una gran variedad de situaciones incluso en instancias que no les son familiares.

Aunado a lo anterior, es importante recordar que los conocimientos del área de ciencias son importantes para el desarrollo de los estudiantes a nivel secundaria en la sociedad; el entender los fenómenos físicos, su interacción con el medio ambiente y la tecnología que deriva de ellos, les ayudan a resolver problemas que se les presenten en su vida cotidiana.

Motivos por los cuales, la presente investigación toma esta área de oportunidad para desarrollar una aplicación de RV enfocada al estudio del movimiento unidimensional, que se imparte durante el proceso de enseñanza – aprendizaje a los alumnos de segundo grado de secundaria, en México.

El presente documento se encuentra estructurado por cuatro capítulos. En el Capítulo 1, se tocan las bases teóricas de la RV. Dentro del Capítulo 2, se describe la metodología que fue utilizada durante el desarrollo de la aplicación de RV, la cual considera una serie de actividades estructuradas, que posibilitan a los diseñadores y desarrolladores ir depurando el producto final en la fase de desarrollo a través de la colaboración y punto de vista de los integrantes del proyecto. Posteriormente en el Capítulo 3, se desarrolló la aplicación de RV a través de las fases de la metodología desde el análisis hasta la evaluación. Y finalmente en el capítulo 4, se mencionan las conclusiones obtenidas.

CAPÍTULO 1

MARCO TEÓRICO

1.1 Realidad Virtual

La RV es definida como un sistema de simulación informática (Zhan, 2011) que tiene la capacidad de recrear situaciones del mundo real o ficticio (Gurău, 2007; Pérez, Zabre & Islas, 2004) con la intención de brindar experiencias reales a sus usuarios (Dickey, 2008), a través de diversos dispositivos inmersivos.

1.2 Clasificación de la RV

Existen varias clasificaciones de la RV, sin embargo, la más utilizada es la mencionada por Shi, 2014; Zhen, 2013; Zhan, 2011; Sala, 2009; Connolly, 2008; Earnshaw, Chilton, & Palmer, 1997, en sus artículos de investigación. En donde dividen la RV dependiendo de su nivel de inmersión en:

1.2.1 RV Inmersiva

La RV Inmersiva (figura 1) se apoya de dispositivos más complejos como por ejemplo, CAVEs, guantes, sensores de movimiento y lentes, con la finalidad de que el usuarios se sumerja completamente dentro de los escenarios virtuales e interactúe con los objetos y personajes a través de su orientación física, movimientos y gestos.



Figura 1. RV inmersiva.

<http://clipset.20minutos.es/wp-content/uploads/2013/11/omni.jpg>

1.2.2 RV no inmersiva o de escritorio

La RV de escritorio utiliza el ordenador personal y estaciones de trabajo. Usando el monitor como un escenario virtual “ventana” e interactuando con él mediante el teclado y el ratón (figura 2). La ventaja de estos sistemas es su facilidad de difusión, debido a que no requiere ninguna adaptación del espacio para que pueda funcionar (Zhan, 2011).



Figura 2. RV no inmersiva.

<http://rehabilitacionymedicinafisica.files.wordpress.com/2009/04/thepitssystemmonitortable100x60cmdatagloves2.jpg>

1.3 Características de la RV

Las características que debe cumplir cualquier aplicación que requiera ser considerada dentro de la RV, según Buerda (1994) son: inmersión, interacción e imaginación. Las cuales forman el “triángulo de la RV” (figura 3).

Triangulo de la realidad virtual

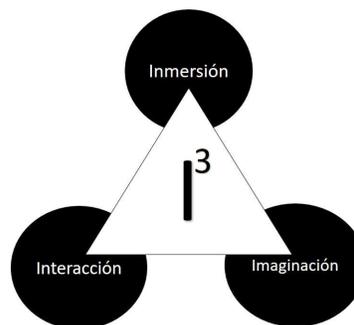


Figura 3. Triangulo de la RV.

1.3.1 Inmersión

Es la característica más relevante de la RV, debido a que le da al usuario la sensación de encontrarse dentro de la situación y ambiente que se le muestra, dando el efecto de que la experiencia obtenida es real (Shi, 2013).

1.3.2 Interacción

La RV debe dar una respuesta en tiempo real ante los actos del usuario (Zhan, 2011), esto quiere decir que el sistema debe reaccionar ante el usuario y sus acciones. No solo presentarse como un mundo estático que solo brinde imágenes y contenido, debe sentirse parte de él y que sus acciones sean parte del desarrollo del mismo.

1.3.3 Imaginación

La RV no es solo una demostración de recursos 3D, también refleja la visión del desarrollador de manera visual (Zhen, 2013), esto quiere decir que el contenido no solo está determinado por el diseño de los elementos, también intervienen la forma en que están integrados y como fueron estructurados por las distintas perspectivas de quienes conforman el proyecto.

1.4 Dispositivos de RV

1.4.1 CAVE

Una CAVE (figura 4) es un entorno generado por computadora que se proyecta alrededor de un usuario para sumergirlo completamente y eliminar toda referencia al mundo real, rastreando y respondiendo a la orientación física, los movimientos y los gestos del usuario.



Figura 4. Uso del C.A.V.E.
<http://ciep.ing.uaslp.mx/mita/wp-content/uploads/2014/06/cave.jpg>

1.4.2 1.4.2 Guantes de RV

Son dispositivos que capturan los movimientos y posición de las manos en el espacio, utilizados por los usuarios para tener interacción con los diferentes objetos dentro de un entorno virtual (figura 5).



Figura 5. Guantes de RV.
<http://www.trendytv.es/wp-content/uploads/2015/09/gloves.jpg>

1.4.3 Lentes de Realidad Virtual

Es un dispositivo que proporciona al usuario la ilusión de estar dentro del entorno virtual, este impacto se logra al combinar los efectos visuales de los lentes junto con la captura del movimiento de su cabeza (figura 6).



Figura 6. Htc lentes de RV.

https://cdn2.cnet.com/img/mSgR-Gen79yo8jvXORrsx2nxX_o=/2015/03/04/e5b30e97-0986-4f43-80e5-a3104894a4bc/htc-vivewhite.jpg

1.4.4 Caminadora de Realidad Virtual

Es un dispositivo que brinda desplazamiento real del usuario dentro de los entornos a través de la detección del movimiento de pies y piernas. Cabe señalar que cualquier movimiento del usuario realizado con la caminadora es replicado en los escenarios de RV (figura 7).



Figura 7. Caminadora de realidad virtual.

<http://www.tecnologiabit.com/wp-content/uploads/2014/04/Virtuix-Omni-Virtual-Reality-Treadmill.jpg>

1.5 Uso de la RV en ambientes educativos

La RV en los ambientes educativos según Pérez (2009), se debe aplicar en actividades didácticas en los casos en donde:

- La enseñanza o el entrenamiento en el mundo real puede ser peligrosa, imposible o inconveniente.
- Pueden suceder errores significativos por parte del alumno o aprendiz en el mundo real. Estos errores pueden ser devastadores o desmoralizadores para el alumno, perjudiciales para el ambiente y/o costosos.
- La interacción con el modelo es igual o más motivadora que la interacción con la situación real.
- La realización de una clase atractiva requiere viajes, dinero y/o logística.
- Se desea lograr experiencias compartidas en un grupo.
- Se desea crear un entorno simulado para lograr los objetivos de aprendizaje.
- Es necesario hacer perceptible lo imperceptible.
- Se quiere desarrollar entornos participativos y de actividades que sólo pueden ser generados por ordenador.
- Las tareas a enseñar requieren destrezas manuales o movimientos físicos.
- Es esencial hacer el aprendizaje más interesante y divertido. La motivación es un factor muy importante, sobre todo en estudiantes que tienen problemas de atención.
- Es necesario proporcionar a una persona discapacitada la oportunidad de realizar experimentos y actividades que de otra manera no podría realizar.

La RV en los ambientes educativos no se debe aplicar en actividades didácticas en los casos en donde:

- Existe otro mecanismo más efectivo en el proceso de enseñanza o aprendizaje de la situación real.
- Es necesaria la interacción con seres humanos reales.

- El entorno virtual puede ser físicamente y/o emocionalmente dañino.
- El entorno virtual puede conducir a un síndrome de “literalización”, en el que el usuario puede confundir el entorno con la realidad.

1.6 La RV y las teorías del aprendizaje

Una de las teorías más beneficiadas por la RV ha sido el constructivismo, ya que en ella se expresa que el alumno debe construir o crear su propio conocimiento y éste tiene especial eficacia cuando se le presenta una motivación por la cual aprender (Valida, 2003).

Papert en 1984, menciona que el constructivismo es una corriente educativa que toma al estudiante como el constructor de su propio conocimiento y como tal necesita materiales para esa construcción y es la cultura la que provee al estudiante estos materiales. Viendo esto habrá diferencias entre aquellos alumnos que tengan mayor acceso a ambientes más ricos e interesantes de aquellos que sean privados de ellos.

Por otro lado, se encuentra también la teoría del conocimiento que plantea que se obtiene una mayor retención cuando el estudiante experimenta directamente el aprendizaje y no solamente lo ve o lo escucha.

La base de esta teoría es el concepto del conocimiento en primera persona, según en el cual un individuo adquiere la mayoría de los conocimientos de su vida diaria mediante experiencias naturales directas, no reflexivas y subjetivas. Al contrario de las técnicas educacionales tradicionales, en las cuales se pone una perspectiva en tercera persona. El conocimiento se transmite a través del maestro resultando indirecto, frecuentemente se abusa del método expositivo, donde el alumno solo se convierte en un ser receptivo y pasivo, llegando a limitar su aprendizaje a un ejercicio de reproducción e ignorando su motivación.

En este sentido, la RV presenta una ventaja, al mantener al estudiante motivado e interesado a seguir aprendiendo y adquirir experiencias que junto con ellas se obtiene el conocimiento (Guillen, 2011). Por ejemplo: el usuario al estar inmerso en un escenario interactivo en el

cual puede desenvolverse con actividades didácticas, le ayuda a crear su propio conocimiento y experiencias.

1.7 Herramientas de desarrollo

Las herramientas de desarrollo son todos aquellos programas y software necesarios para crear cualquier aplicación y sus elementos (en este caso enfocadas a la RV). Esto abarca desde las herramientas de modelado, donde se crean todos los objetos y estructuras bidimensionales y tridimensionales, pasando por las herramientas de texturización, animación y audio, hasta el motor gráfico de representación donde se integran cada uno de los elementos desarrollados para dar como resultado una aplicación de RV funcional.

1.7.1 Herramientas de texturizado

Son editores de gráficos e imágenes, permite tratar la imagen en segmentos mediante capas, para poder modificar los elementos de la imagen de manera independiente. La imagen resultante se puede grabar en diversos formatos, dependiendo la necesidad y compatibilidad con el motor gráfico.

1.7.1.1 Adobe Photoshop

Software dedicado a la edición de gráficos 2D, con calidad fotográfica y de alta resolución. Cuenta con gran cantidad de herramientas, para edición y retoque de imágenes. Es líder de mercado en aplicaciones de edición de imágenes (figura 8).

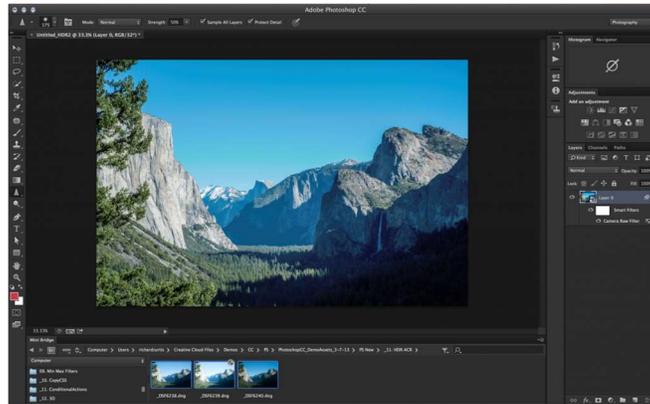


Figura 8. Fotoshop

1.7.2 Herramientas de modelado

Es software altamente especializado que generan modelos 3D de objetos, personajes y arquitecturas, otorgando una percepción de alta calidad sobre su creación y presentación. Algunos ejemplos de estas herramientas son: Maya, Solid Works, 3Dstudio max, Blender, etc.

1.7.2.1 Autodesk Maya

Autodesk Maya (figura 9) es un software dedicado al desarrollo de gráficos 3D, efectos visuales y animación. Tiene una plataforma amplia en herramientas (Scale, Rotate, Move, Extrude, Uv projector) de edición para proyectos de renderizado, desarrollo de videojuegos y composiciones 3D. Es capaz de trabajar en cualquier tipo de superficie (Nurbs, Poligonos, Subdivisiones) lo cual la convierte en uno de los programas más versátiles del mercado 3D.

Requisitos de instalación:

- Microsoft® Windows® 10 Professional, Windows® 8.1 Professional, and Windows® 7 (SP1) operating system o en Apple® Mac OS® X 10.9.5, 10.10.x, and 10.11.3 como sistema operativo.
- 8 GB en memoria RAM.

- 4 GB en disco duro libres.
- 2 GB en GPU (Tarjeta de video) recomendado.

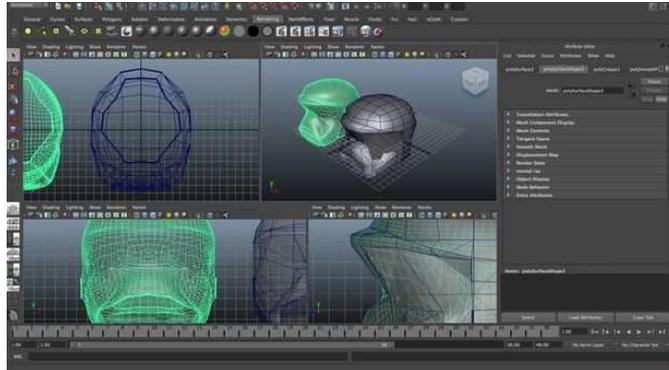


Figura 9. Interface Autodesk Maya

Cabe señalar que se eligió esta herramienta por la licencia de educación que ofrece, en la cual se especifica que se puede usar el software de manera completa mientras el producto que se desarrolle no tenga fines lucrativos.

1.7.3 Herramientas de animación

Son programas específicos de modelado que con el uso de herramientas y complementos (Plugins) que permiten modificar los modelos ya creados, con el objetivo de otorgarles movimiento en una línea de tiempo específica.

1.7.3.1 Daz 3D

Software de animación y renderizado que utiliza la plataforma “Génesis”, que brinda la capacidad de estandarizar la anatomía de los personajes, utilizando los huesos y escala de la misma manera en la mayoría de los casos, exceptuando los animales. Dando como resultado una poderosa herramienta de animación para videojuegos (figura 10).

Requisitos de instalación:

- Windows 7, 8, 10 o Mac OS X 10.6 o superior.
- Intel Core i5 (o equivalente).
- 4 GB en memoria RAM.
- 1 GB de espacio libre en disco duro (No incluye assets).
- 4 GB en GPU.

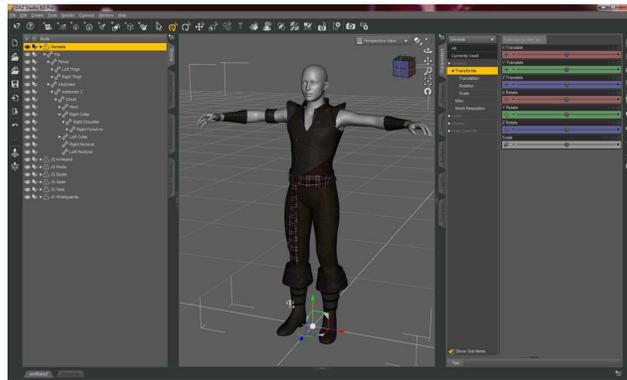


Figura 10. Interface Daz 3D

1.7.3.2 Iclone

Es un software de representación, renderización, animación 3D en tiempo real. Que puede crear animaciones a modo de película. Brinda la capacidad de generar secuencias de alta calidad debido a que se pueden controlar cada uno de los movimientos de las animaciones (faciales y/o esqueléticas) para el cuerpo. Hoy en día se ha convertido en una gran herramienta para los desarrolladores (figura 11).

Requisitos de instalación:

- Windows 7, 8 y 10 (64 bits).
- Procesador: Dual Core CPU o mayor.
- 2GB en RAM.

- 2GB de espacio libre en Disco Duro.
- Pantalla con resolución: 1024 x 768.
- GPU: 512 MB, NVIDIA GeForce 8 Series o mayor / ATI HD 3000 Series o mayor.



Figura 11. Interface Iclone

1.7.3.3 Iclone3DXchange

Es un software robusto que facilita la identificación y conversión de los diversos formatos de personajes 3D existentes para su futuro uso en el software Iclone, además ayuda a identificar las diversas estructuras de esqueleto y faciales necesarias para crear animaciones. Una vez terminado el proceso de animación ayuda en la exportación de los personajes integrándolos en un solo archivo el formato necesario para que el motor gráfico de representación lo interprete (figura 12).

Requisitos de instalación:

- Windows 7, 8, 10 (64 bits).
- Procesador: Dual Core CPU o mayor.
- 2GB en RAM.
- 2GB de espacio libre en Disco Duro.
- Pantalla con resolución: 1024 x 768.
- GPU: 512 MB, NVIDIA GeForce 8 Series o mayor / ATI HD 3000 Series o mayor.

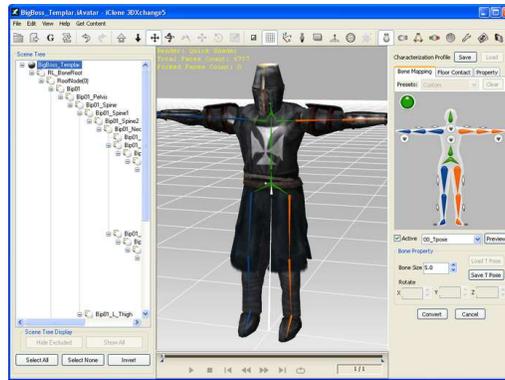


Figura 12. Interface Iclone 3DXange

1.7.4 Adobe Audition

Es una aplicación dedicada a la creación, grabación y edición de pistas de audio digital con una gran cantidad de herramienta de refinación de audio. Dentro de ella se pueden crear diversos efectos para instrumentos musicales, voz o sonidos de ambientación los cuales pueden ser exportados en diversos formatos dependiendo el motor gráfico de representación que se esté utilizando (figura 13).

Requisitos:

- Windows 7, 8, 10 o Mac OS X 10.6 o superior
- Intel Core i5 (o equivalente).
- 4 GB en memoria RAM.
- 4 GB de espacio libre en disco duro.
- Tarjeta de sonido compatible con el protocolo ASIO, WASAPI o Microsoft WDM/MME.
- Conexión a internet.



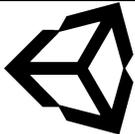
Figura 13. Interface Audacity

1.7.5 Motores gráficos

Los motores gráficos son software que representan gráficos 3D y soportan las diferentes rutinas de programación de un videojuego o entorno virtual. Esto se logra debido a que utilizando un motor de renderizado (gráficos y animación), un motor físico (detección de colisiones y ambiente) inteligencia artificial, un administrador de memoria y de ser necesario conexión con un servidor.

A continuación, se muestra una tabla comparativa de algunos motores gráficos de representación:

Tabla 1. Tabla de comparación de motores gráficos de representación

Motores gráficos de representación			
Características técnicas y empresariales	Unity	Cry Engine	Unreal
Descripción	 Motor de video juego multi plataforma estructurado en módulos (scripts, animaciones y pantallas de operadores). Lo que lo hace fácil de aprender y utilizar, posicionándose como uno de los motores más famosos en la actualidad.	 Motor de representación originalmente usado para la empresa Nvidia, soportado por una gran cantidad de plataformas. En la actualidad es uno de los motores más potentes del mercado, para uso profesional.	 Motor gráfico, compatible con Opengl y Directx, actualmente es considerado uno de los mejores motores de videojuegos utilizado en varias plataformas y para trabajos independientes.
Licencia	Libre (Plus) 35/ mes (Pro) 125 / mes	50 / mes 150 / mes	Libre Pago del 5% de todas las regalías
Plataforma	Windows y OSX	Windows y OSX	Windows y OSX
Lenguaje de programación	C#, Javascript y Boo	C++	C++ y Boo
Soporte de modelado	3ds Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D y Cheetah3D	3ds Max, Maya, Blender y CryBlend	3ds Max, Maya y Blender
Plataforma	Windows, OSX, Linux, Web, Android, IOS, Play station, Xbox y Nintendo	Windows, OSX, Web, Android, IOS, Play station, Xbox y Nintendo	Windows, OSX, Web, Android, IOS, Play station y Xbox
Soporte	Comunidad grande, tutoriales, soporte técnico y foros	Comunidad media, Tutoriales, foros, soporte técnico, cursos webinarios (Incluidos en la compra de licencia)	Comunidad grande, tutoriales, soporte técnico, foros
Tienda	Recursos y plugins Unity marketplace	Recursos y plugins Crytek marketplace	Recursos y plugins Unreal Marketplace
Curva de aprendizaje	Normal	Compleja	Normal

Después de analizar las características de cada uno de los motores de representación gráfica y tomando en cuenta las características aplicación de RV que se va a desarrollar, se llegó a la conclusión de utilizar Unity3D debido a que:

- Presenta tanta compatibilidad con Windows y OSX que son los Sistemas Operativos con los que trabajan las computadoras que serán utilizadas durante toda la creación de este proyecto.
- Es compatible con las siguientes herramientas de modelado y animación: Maya, Daz 3D, IClone y 3DXchange, en las cuales se crearán, optimizaran y animaran los componentes 3D (objetos, estructuras y personajes).
- Maneja una versatilidad de lenguajes de programación, lo cual va a permitir trabajar e integrar varios scripts de diferentes lenguajes. Ya que se integrará el dispositivo Leap Motion, el cual utiliza C# de manera nativa y además se tiene pensado utilizar scripts en lenguaje Java para la programación de animaciones, máquinas de estado, alteración de propiedades (cámaras, gravedad de los entornos, creación de objetos, etc.), reciclaje de objetos y control de framerate.
- Y además brinda soporte en línea, cuenta con comunidades, foros grandes y una tienda de objetos (assets store) con acceso a contenido gratuito.

CAPÍTULO 2

METODOLOGÍA DE DESARROLLO

A continuación, se describen las fases y actividades correspondientes a la metodología de desarrollo evolutivo para escenarios tridimensionales (Morales, et al., 2015) (figura 14), que será utilizada para el desarrollo del presente proyecto.

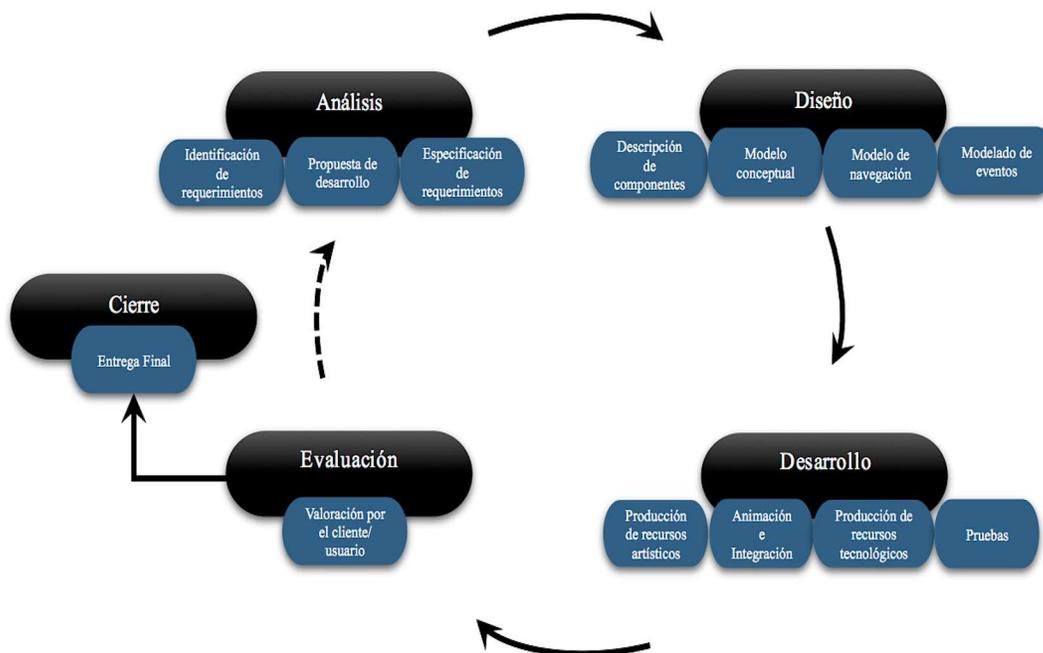


Figura 14. Metodología de desarrollo evolutivo para escenarios tridimensionales (Morales, et. al, 2015)

2.1 Fase de Análisis

La finalidad de esta fase, es tener una perspectiva clara del contexto en el cual se va a enfocar la aplicación de RV. Para lo cual, es necesario definir y comprender las necesidades del cliente a través de tres actividades: identificación de requerimientos, propuesta de desarrollo y especificación de requerimientos.

2.1.1 Identificación de requerimientos

Se analiza el caso planteado por el cliente para visualizar las necesidades que deberán ser cubiertas.

2.1.2 Propuesta de desarrollo

En esta actividad, se forjan diferentes planes factibles de desarrollo que cubran las necesidades del cliente y cumplan con las restricciones establecidas de presupuesto y tiempo.

2.1.3 Especificación de requerimientos

Con base en la información obtenida en las actividades anteriores, se crea la documentación que especifica de forma clara cuáles serán los alcances, atributos, requerimientos y limitantes del producto a desarrollar, es decir, los requerimientos funcionales y no funcionales del producto.

Tabla 2. Tabla de especificaciones.

Descripción general del producto:			
Objetivo:			
Requerimientos			
No.	Descripción	Funcional	No funcional
1			

2.2 Fases de Diseño

En esta fase, se crean los modelos conceptuales y se definen las características explícitas e implícitas de cada uno de los componentes (espacios arquitectónicos, objetos, personajes, escenarios, secuencias de audio y video) que conforman la aplicación.

2.2.1 Modelado conceptual

Partiendo de la información generada en la fase anterior, el cliente en conjunto con el equipo de diseñadores crea el concepto (modelos o bocetos) que serán la base para desarrollar los escenarios tridimensionales.

2.2.2 Modelado de Navegación

Aquí se define la forma en como el usuario podrá recorrer e interactuar con los elementos que componen la visita virtual, así como la técnica de representación que se utilizará (tipo de vista del usuario: primera persona, segunda persona o tercera persona).

2.2.3 Descripción de componentes

Su propósito es hacer una descripción detallada de los elementos que integraran la aplicación de RV para identificar sus características físicas particulares, el tipo de interactividad y su comportamiento dentro de los escenarios. Para ello, se recomienda el uso de tablas de descripción de componentes, las cuales servirán para que los modeladores tridimensionales puedan revisar de manera inmediata los principales atributos que conforman a cada elemento (Tabla 3).

Tabla 3. Descripción de componentes

Identificación:	Tipo de plantilla:	Descripción de objetos de ambientación					
	Objeto:	Características:				Cantidad:	Escenario:
		Color	Forma	Textura	Medidas		
O-001							
O-002							

Nota: O=objeto; conviene distinguir que solo se describe la tabla de objetos de ambientación, sin embargo, cada tipo de componente tiene sus propios atributos.

2.2.4 Modelado de Eventos

En esta actividad se detallan los sucesos que serán detectados dentro del sistema o aplicación RV y las acciones correspondientes que los usuarios, los objetos, y/o entornos deberán realizar. Es prudente advertir que si se desea un alto nivel de inmersión el modelo de eventos será más complejo.

2.3 Fase de Desarrollo

El objetivo de esta fase es construir el producto. Para ello, se trabaja de forma iterativa e incremental para lograr una versión ejecutable del sistemas o aplicación de RV, al finalizar cada iteración, cabe resaltar, que si los elementos desarrollados no cumplen con los criterios establecidos se podrá retornar a las actividades anteriores con el objetivo de modificar los errores.

La fase de desarrollo deberá seguir una secuencia de actividades que se muestran en la figura 15 y se puntualizan a continuación:

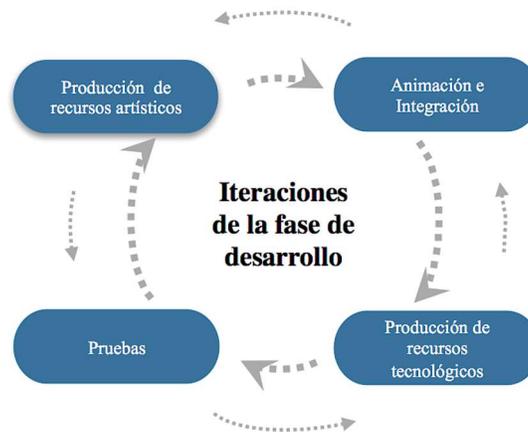


Figura 15. Fase de desarrollo

2.3.1 Producción de recursos artísticos

Con el uso de herramientas de software para modelado tridimensional se crean los espacios arquitectónicos, objetos, personajes y terrenos que integran los escenarios del sistema o aplicación RV, así como los materiales y texturas que serán aplicados a los componentes tridimensionales a fin de aumentar su realismo. De igual modo, se desarrollan las secuencias de audio y video necesarias para complementar la representación de la información deseada.

2.3.2 Animación e Integración

En lo que respecta a la actividad de animación se producen secuencias de movimiento, cambio de posición o transformación de parámetros de los componentes según los modelos de navegación y eventos. Posteriormente, en la etapa de integración todos los componentes tridimensionales, secuencias de audio y video son incorporados en los escenarios o ambientes virtuales.

2.3.3 Producción de recursos tecnológicos

Con la información proveniente de la etapa de diseño, se codifican y/o editan los eventos que disparan las acciones de respuesta establecidas para cada uno de ellos.

2.3.4 Pruebas

Una vez finalizada cada iteración, se verifica el funcionamiento y la integración de los componentes tridimensionales con la finalidad de obtener información útil, en aras de mejorar el producto en aquellos aspectos concernientes a los parámetros funcionales que fueron establecidos en las fases de análisis y diseño.

2.4 Fase de Evaluación

Valoración por el cliente/usuario. Cuando se considera que los componentes desarrollados se han integrado de manera adecuada en cada escenario y que éstos conforman de manera coherente la representación virtual del espacio real establecido, el producto es considerado como versión Alfa, la cual debe ser puesta a consideración del cliente a fin de confirmar que cumpla con los requerimientos planteados al inicio del proyecto.

Finalmente, la retroalimentación del cliente dará como resultado una versión Beta del producto, que se pone a consideración del usuario con pocos conocimientos acerca de la navegación en entornos virtuales, para valorar algún fallo no conocido por el equipo de desarrollo; con la finalidad de que el producto pueda ser liberado.

2.5 Fase de Cierre y Entrega Final

En esta fase se entrega la versión final del sistema o aplicación de RV, lista para implementarse en un ambiente con usuarios y condiciones reales, la cual puede ser

susceptible de recibir un plan de mantenimiento y/o actualización de contenidos o componentes según los acuerdos trazados con el cliente.

CAPÍTULO 3

IMPLEMENTACIÓN DE LA METODOLOGÍA

3.1 Fase de Análisis

En esta fase se busca tener una perspectiva clara de la aplicación educativa que se va a desarrollar. Para lo cual, fue necesario definir: los requerimientos, la propuesta de desarrollo y la especificación de requerimientos.

3.1.1 Identificación de requerimientos

De acuerdo al área de oportunidad encontrada en la enseñanza de las ciencias en México descrita en la introducción, se decidió analizar el programa de la materia de ciencias que imparte la SEP en segundo año de secundaria (tabla 4), con el objetivo de elegir un tema que pudiera ser representado con la tecnología de RV.

Después de hacer la revisión se llegó a la conclusión de elegir el tema percepción del movimiento y el trabajo de Galileo Galilei, ya que son temas que son difíciles de representar dentro del salón de clases, por ejemplo, resulta inconveniente y peligroso tirar objetos de diversas características físicas desde un segundo piso para percibir el efecto de la caída libre.

Tabla 4.Extracto de contenido del programa de la materia ciencias del segundo año de secundaria

Propósito del bloque	Tema	Subtema	Habilidades de pensamiento
<p>Que los alumnos:</p> <p>1. Analice y comprenda los conceptos básicos del movimiento y sus relaciones, lo describan e interpreten mediante algunas formas de representación simbólica y gráfica.</p> <p>2. Valoren las repercusiones de los trabajos de Galileo acerca de la caída libre en el desarrollo de la física en especial en la forma de analizar el movimiento.</p> <p>3. Apliquen e integren habilidades actitudes y valores durante el desarrollo de proyectos enfatizando el diseño y la realización de los experimentos, que les permita relacionar los conceptos estudiados con los fenómenos y su entorno.</p> <p>4. Reflexionen acerca de las implicaciones sociales de algunos desarrollos tecnológicos relacionados con la medición del movimiento</p>	La percepción de movimiento	¿Cómo sabemos que algo se mueve?	Reconocer, comparar, relacionar y describir
		¿Cómo describimos el movimiento de los objetos?	Describir, comparar, interpretar, identificar y construir
		Un tipo particular de movimiento: el movimiento ondulatorio	Aplicar, describir, diferenciar, utilizar, explicar
	El trabajo de Galileo Galilei. Una aportación importante para la ciencia	¿Cómo es el movimiento de los objetos que caen?	Identificar, aplicar, describir, constatar, validar y analizar
		¿Cómo es el movimiento cuando la velocidad cambia? Aceleración	Aplicar, describir, identificar, diferenciar, e interpretar
	Integración y aplicación	<p>¿Cómo se propagan y previenen los terremotos?,</p> <p>¿Cómo se mide la velocidad en los deportes?,</p> <p>¿Cómo potenciar nuestros sentidos para saber más?</p>	Elaborar, representar, interpretar, expresar, diseñar, comunicar, describir, manifestar, analizar, discutir.

3.1.2 Propuesta de desarrollo

Por las ventajas pedagógicas que ofrece la RV como son: aumentan la retención, impulsan el aprendizaje conceptual y procedimental, motivan al usuario a seguir aprendiendo, etcétera dentro del ámbito de la educación, se propuso crear una aplicación de RV didáctica que sirva como una herramienta de apoyo a la experimentación y comprensión del concepto de caída libre y su historia.

La cual se planteó que estuviera integrada por seis escenarios; plaza de pisa, mirado de la torre, la luna, marte, júpiter y la cámara de vacío de la nasa. En el escenario de la plaza de pisa se dará la bienvenida al usuario (tabla 5) y se representará a través una breve explicación la evolución y el concepto acerca del fenómeno de caída libre, las cuales serán expuestas por la representación en 3D del Astrónomo Galileo Galilei considerado actualmente como el

padre de la física moderna por sus descubrimientos de la teoría moderna de la caída libre que refuta la teoría de Aristóteles.

Tabla 5. Guion Bienvenida

Id de Dialogo	Personaje	Diálogo
DI1	Galileo Galilei	Hey hey tu
DI2		Parces non essere il postro per te (no pareces ser de este lugar)
DI3		¿Non capisci che non parlo italiano? (¿no me entiendes no hablas italiano?)
	Usuario
DI4	Galileo Galilei	Parle francais? ¿Non?
	Usuario	No entiendo
DI5	Galileo Galilei	O!! Así que hablas español?
	Usuario	Si!!
DI6	Galileo Galilei	¿Hola soy Galileo no pareces ser de por aquí dime que haces? Ummmmmm, A si que también te preguntas que cae primero.
DI7		Bueno al igual que tú y yo, muchas personas se han hecho esta pregunta.
DI8		¡Pero yo el gran Galileo! ¡Junto contigo claro ! le daremos respuesta.
DI9		Claro que en mi tiempo hay muchas limitaciones, así que debo arreglar unas tantas cosas antes.
DI10		Anda regresa a tu tiempo y utiliza esos raros azulejos para comprobarlo. ¡Nos veremos Amico!!!!

En los cinco escenarios restantes (tabla 6,7,8,9,10) se planea que antes de comenzar a experimentar se le dé al usuario una breve explicación para ubicarlo en el contexto, cabe resaltar que el estudiante podrá experimentar y observar el comportamiento del fenómeno de caída libre lanzando objetos con diferentes características (forma, peso y tamaño). De esta manera se pretende que, a través de estas experiencias los estudiantes puedan alcanzar un aprendizaje más significativo.

Tabla 6. Guión torre

Id de dialogo	Personaje	Diálogo
DT1	Galileo Galilei	Que bella vista no, estamos en el observatorio de la torre aquí podrás experimentar, y tirar objetos por el balcón.
	Usuario	- -
DT2	Galileo Galilei	Bien ahora vamos hacer esto más interesante, por estos portales podrás viajar a diferentes lugares anda escoge uno.

Tabla 7. Guión luna

Id de dialogo	Personaje	Diálogo
DL1	Galileo Galilei	La luna el satélite natural de nuestro planeta, es ochenta veces más pequeña que la tierra sin atmosfera y con una fracción

		de la gravedad de la tierra aquí veras que los objetos caen con mayor lentitud.
DL2	Galileo Galilei	Mira que majestuosa se ve la tierra a mi parecer el planeta más hermoso que existe.
	Usuario	- -
DL3	Galileo Galilei	Anda escoge otro portal veamos adonde nos lleva.

Tabla 8. Guión marte

Id de dialogo	Personaje	Diálogo
DM1	Galileo Galilei	(tose) Marte, o conocido como el planeta rojo es muy similar a la tierra en algunos aspectos aquí podrás probar la resistencia al aire con una gravedad menor a la dela tierra.
	Usuario	- -
DM2	Galileo Galilei	Vamos más lugares nos esperan y en particular no me gusta el polvo.

Tabla 9. Guión júpiter

Id de dialogo	Personaje	Diálogo
DJ1	Galileo Galilei	Júpiter el planeta con la mayor gravedad del sistema solar, es 318 veces más grande que la tierra, a diferencia de esta no tiene superficie tangible ya que es de gas, aún hoy, mucho de lo que pasa de bajo de esas gigantescas nubes de hidrogeno es un misterio.
	Usuario	- -
DJ2	Galileo Galilei	Anda escoge otro portal esto de flotar entre nubes me pone nervioso.

Tabla 10. Guión cámara de vacío

	Personaje	Diálogo
DV1	Galileo Galilei	Esta es la cámara de vacío más grande de tu era en mi tiempo no existía ni una jajaj. Aquí probaras la caída libre sin resistencia al aire.
	Usuario	- -
DV2	Galileo Galilei	Toma otro portal no resisto la respiración mas.

Nota: Los espacios vacíos (--) son pausas que realizará el sistema entre un diálogo y otro.

3.1.3 Especificación de requerimientos

En base a la investigación realizada en diversos artículos científicos en los cuales se utilizan sistemas o aplicaciones de la RV en el ámbito educativo (Shi, 2004; Zhan, 2011; Ibañes et al., 2010; Kaufmann, & Schmalstieg, 2006; Dickey, 2005; Rahim, 2014; Antonietti, 2000) y al área de oportunidad que se presenta en México como ya se menciono, se creó una tabla en donde se definen los requerimientos funcionales y no funcionales, lo anterior, con el propósito de que se pudiera obtener una aplicación didáctica de RV utilizable (tabla 10).

Tabla 11. Requisitos funcionales y no funcionales

Descripción general del producto:	Aplicación de RV para el estudio del movimiento unidimensional: una actividad didáctica.		
Objetivo:	Crear una aplicación de RV, enfocada al estudio del movimiento unidimensional, que puede ser utilizada como material didáctico en el proceso de enseñanza aprendizaje en el tema de fenómeno de caída libre.		
Requerimientos			
No.	Descripción	Funcional	No funcional
1	La aplicación de RV debe contar con características que le permitan al usuario acceder a él, desde cualquier equipo de cómputo.		X
2	La aplicación debe ser lo suficiente intuitiva e inmersiva para retener la atención del usuario, pero no de manera excesiva para distraerlo.	X	
3	Debe proporcionar una interacción cercana y amigable al usuario.	X	
4	El fenómeno debe ser explicado con cada una de sus variables y sus cambios.	X	
5	El personaje principal (Galileo Galilei) mencionara la evolución y concepto del fenómeno de caída libre mediante una animación.	X	
6	Explicar a través de una animación como Galileo Galilei supero las limitaciones tecnológicas que enfrente al querer comprobar su teoría.	X	
7	Los entornos de RV para la experimentación contaran con diferentes tipos de gravedad y resistencia al aire.	X	
8	Durante la etapa de experimentación el usuario debe tendrá que notar los cambios entre escenarios.		X
9	Los objetos tridimensionales deben estar representados lo más parecidos al mundo real.	X	
10	Los personajes de ambientación deben tener tanto características físicas como vestimenta de la época renacentista.	X	
11	El personaje principal debe contar con características físicas similares a las de Galileo Galilei durante sus primeros años en que desarrollo su investigación acerca del fenómeno de caída libre.	X	
12	Las animaciones de los personajes deben ser detalladas y de calidad para que el usuario observe movimientos naturales.	X	
13	El personaje principal debe reaccionar a las respuestas del usuario.	X	
14	Las texturas de los objetos deben ser lo más reales posibles.	X	
15	Los entornos de experimentación deben ser semejantes a los lugares reales: Plaza de Pisa, Luna, Marte, Júpiter y Cámara de vacío.	X	
16	El usuario puede decidir qué rumbo tomar dentro de cada escenario.	X	
17	La vista del usuario será con perspectiva de primera persona.	X	
18	Los usuarios podrán tomar los objetos y dejarlos caer cuando lo consideren conveniente durante la experimentación.	X	

3.2 Fase de Diseño

En esta fase, se crearon los modelos conceptuales y se especificaron las características de cada uno de los componentes (espacios arquitectónicos, objetos, personajes, escenarios, guiones, dinámicas, secuencias de audio y video), que conformaran la aplicación.

3.2.1 Modelado conceptual

Después de haber completado la fase de análisis, se inició con la elaboración de los mapas de distribución seguido del escenario de inicio (figura 16 y 17) y de los escenarios de experimentación los cuales se observan en las siguientes figuras 16 a la 23.

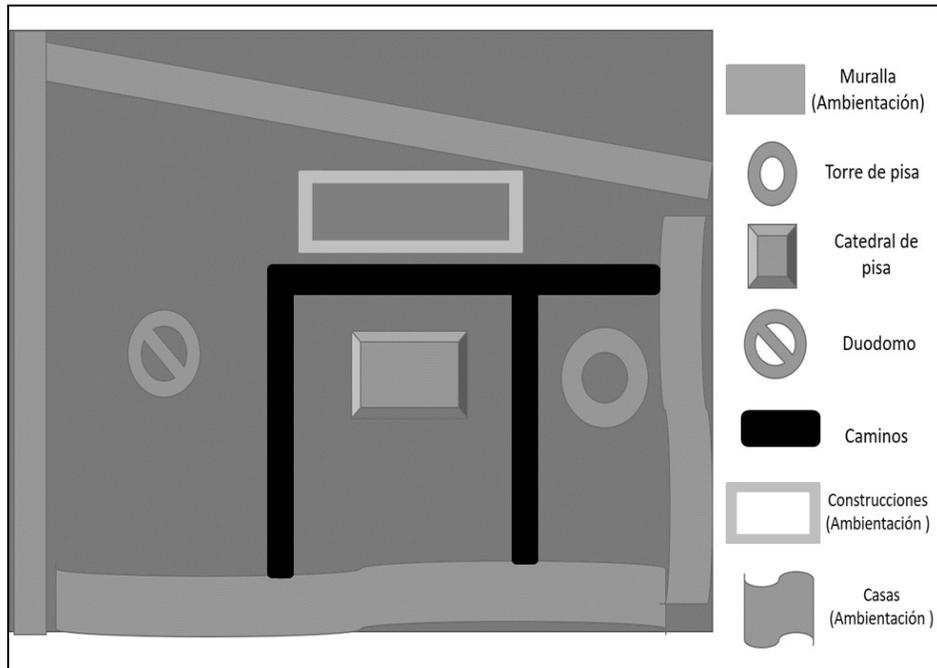


Figura 16. Mapa distribución general para el escenario de la plaza de Pisa

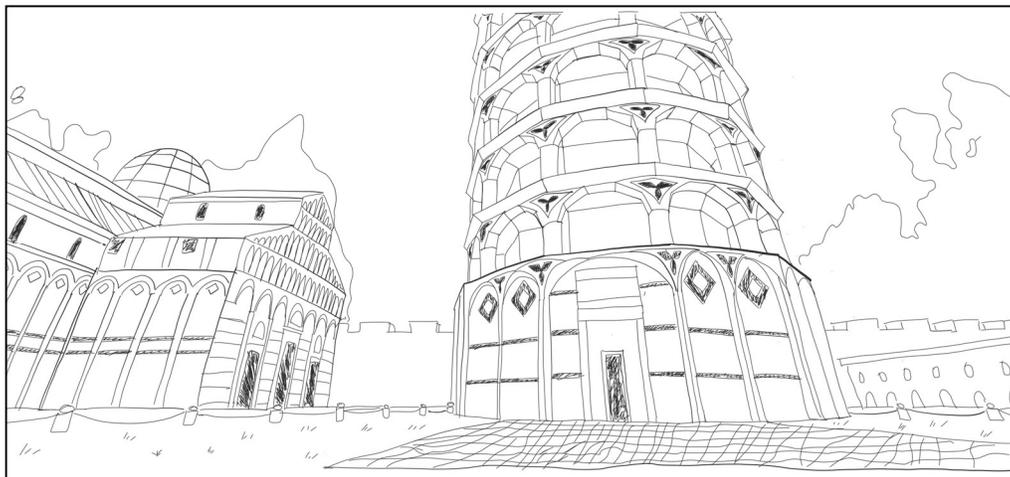


Figura 17. Bosquejo general de distribución de la plaza de Pisa

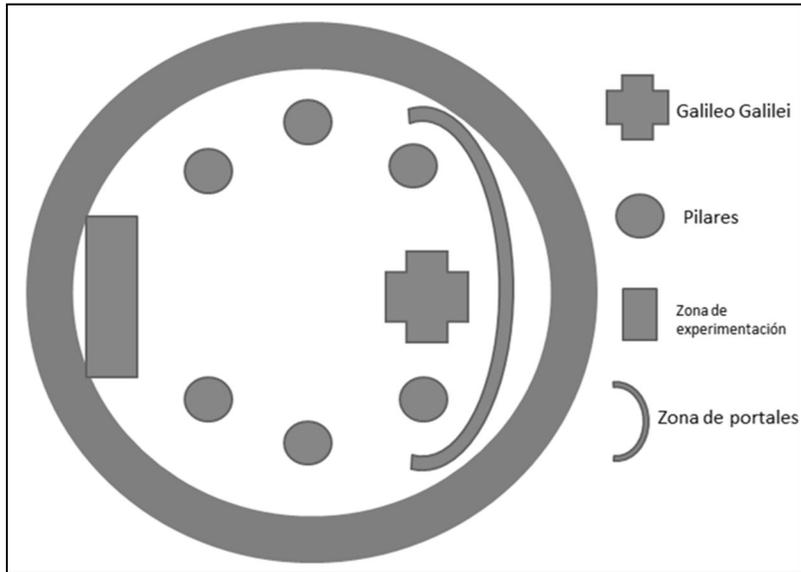


Figura 18. Mapa distribución general de los escenarios de experimentación (Torre de pisa, Jupiter, Marte, Luna y Cámara de vacío)

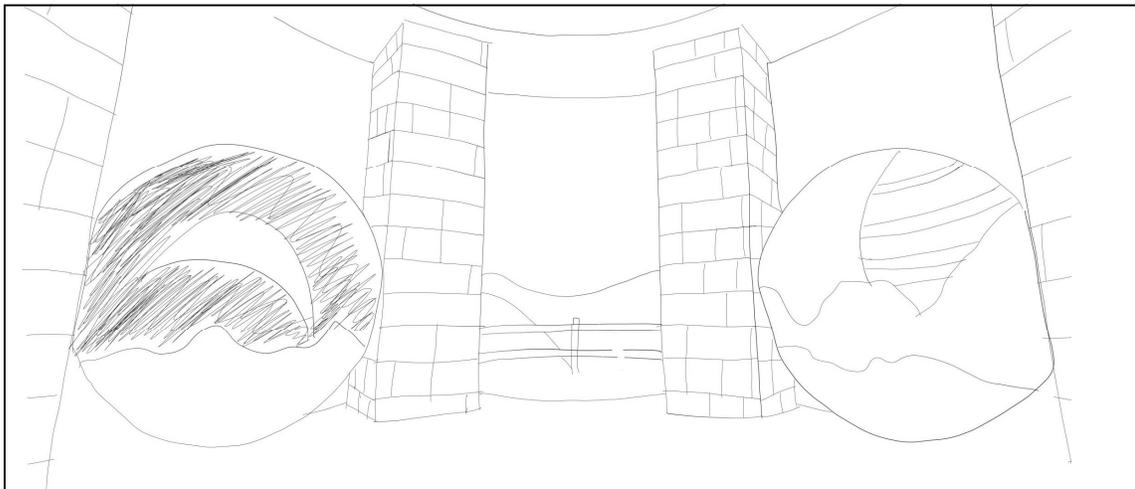


Figura 19. Diseño de la vista del mirador de la torre de pisa desde el planta tierra

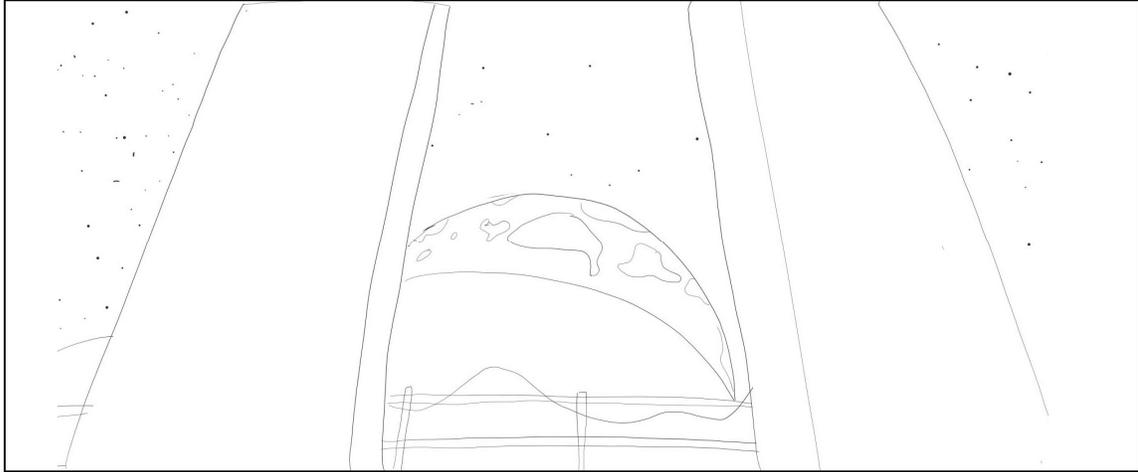


Figura 20. Diseño de vista del mirador de la torre desde la Luna

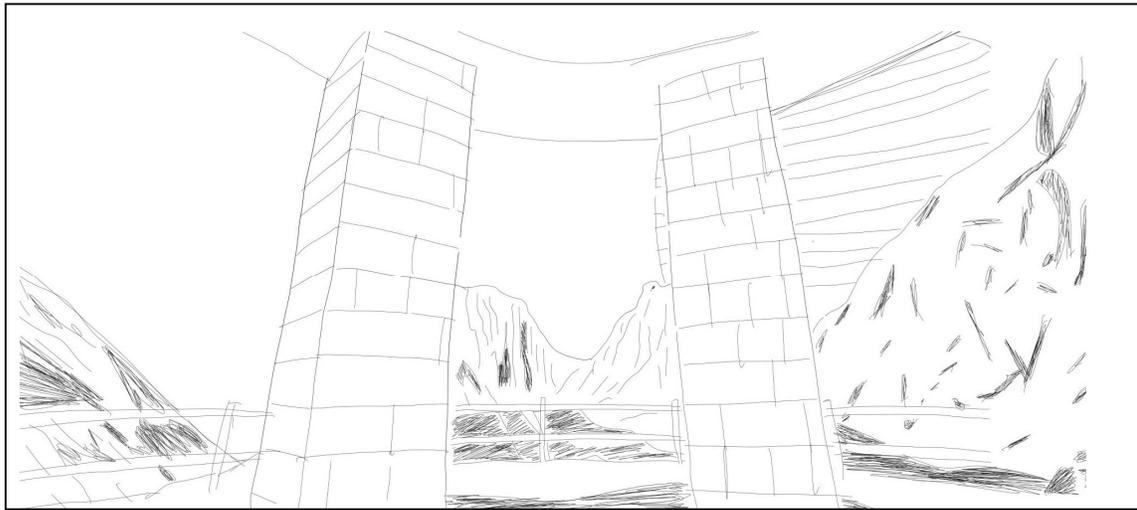


Figura 21. Diseño de vista del mirador de la torre desde Marte

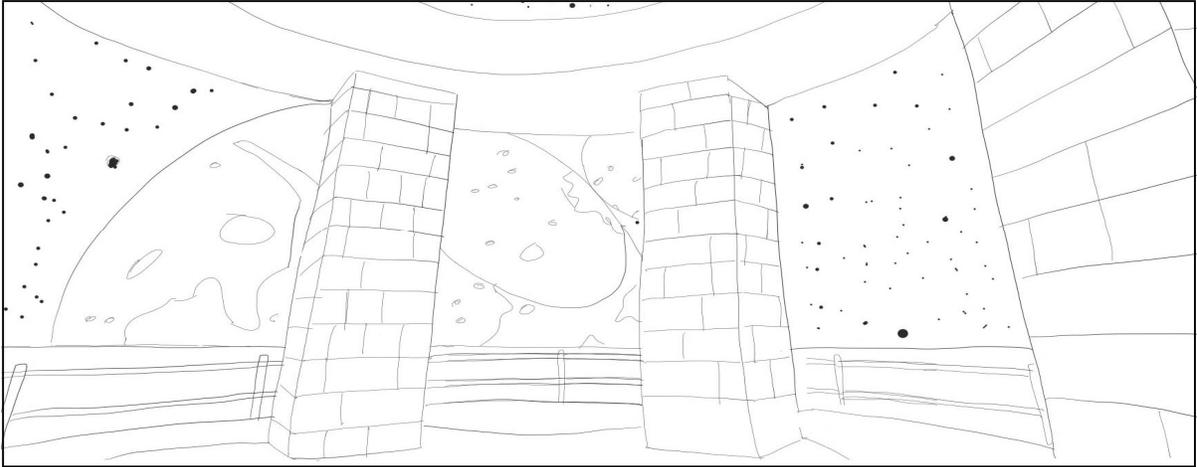


Figura 22. Diseño de vista del mirador de la torre desde Júpiter

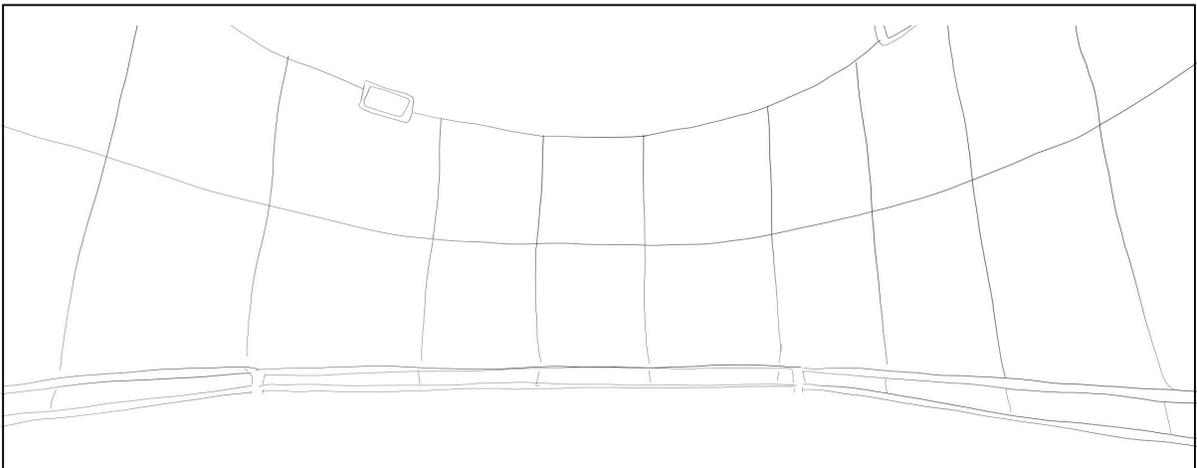


Figura 23. Diseño de vista de la Cámara de vacío de la NASA

Nota: Cabe señalar que estos diseños 2D se realizaron utilizando imágenes reales como referencia a través del programa Adobe Draw.

Para poder crear los modelos 3D en la fase de desarrollo con dimensiones reales se buscaron en la web los planos con medidas tomándose como referencia las siguientes figuras 24 y 25.

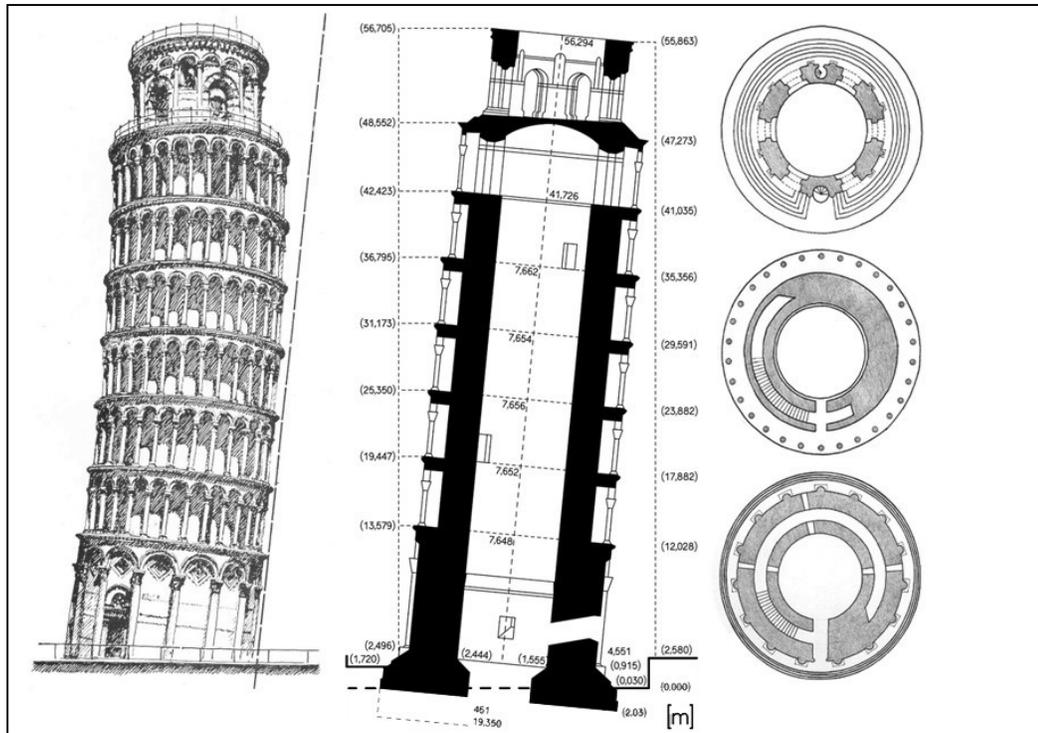


Figura 24. Planos de torres de pisa

http://4.bp.blogspot.com/-BJCgTdEVlfi/T1ajtqWkvTI/AAAAAAAAA9Q/M_HJ8dzwLU/s1600/Seccion+por+el+plano+maximo+de+inclinacion+de+Pisa.jpg

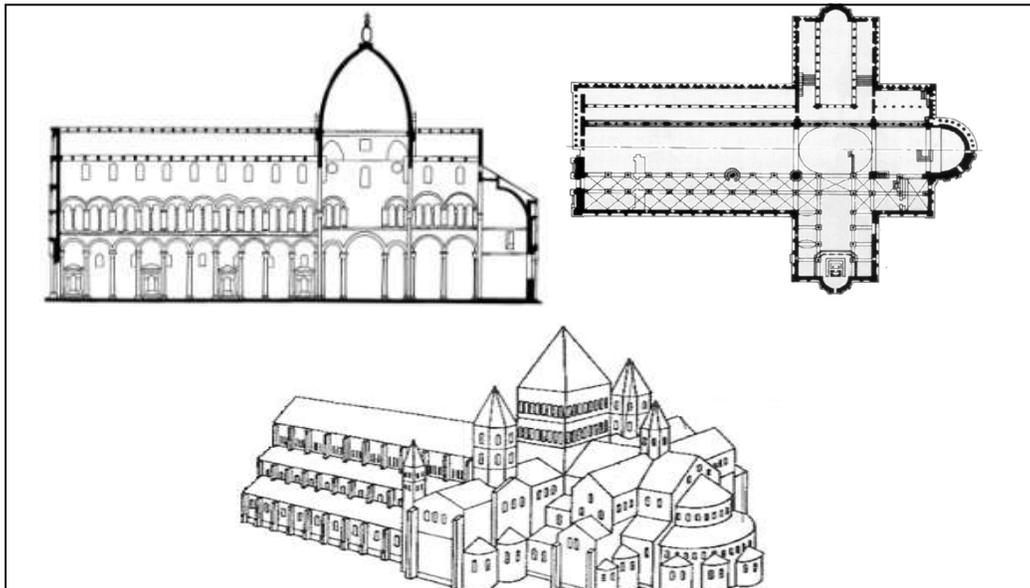


Figura 25. Planos de catedral de pisa

<https://arquistoria.files.wordpress.com/2012/10/pisa-4.jpg>

A demás en esta fase con el objetivo de lograr un mayor realismo en las estructuras y objetos que conformaran la aplicación, se decidió buscar en internet, en google maps y en el sitio de la NASA diversas imágenes (figura 26) que servirán como texturas de los modelos 3D a desarrollar.



Figura 26. Torre de pisa desde Streetview

3.2.2 Modelado de navegación

En este apartado se creó el diagrama de navegación (figura 27) que muestra el recorrido que podrá realizar el usuario cuando utilice la aplicación y experimente el fenómeno de caída libre (sub rutina) en los diferentes escenarios (figura 28).

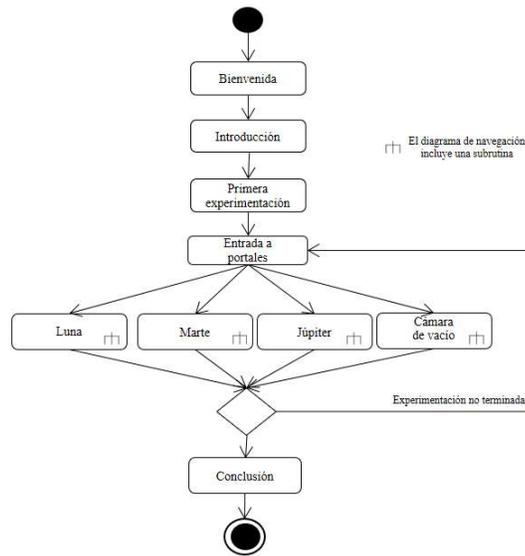


Figura 27. Diagrama de navegación de la aplicación

Nota: La ejecución de la sub rutina será la misma en todos los escenarios.

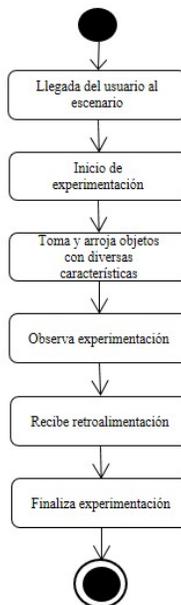


Figura 28. Diagrama de navegación de subrutina de experimentación

Cuando el usuario ingresa a la aplicación didáctica el personaje de Galileo lo recibirá y le presenta una introducción donde se le explican varios aspectos de la caída libre, una vez terminada la secuencia de animación, el usuario podrá subir al observatorio de la torre de pisa y a la primera experimentación. Después de esto podrá ir a cualquiera de los 4 escenarios donde podrá experimentar el fenómeno, con diferentes condiciones de gravedad y resistencia al aire. Después de pasar por todos los escenarios se le presentara una conclusión acerca de los experimentos desarrollados por Galileo Galilei.

Durante el recorrido dentro de la aplicación el usuario se encontrará en una vista de primera persona (figura 29), lo que le ayudará a tener una mejor perspectiva al utilizar el dispositivo Leap Motion y una experiencia más personal en el entorno.



Figura 29. Perspectiva en primera persona leap.

3.2.3 Story Board

Para crear la representación de la cinemática que se pretende incluir como bienvenida e introducción dentro del escenario de la plaza de pisa se creó un storyboard (tabla 12). Considerado esté como una representación visual, que crea una idea general de la animación, especifica movimientos, diálogos y cambios de cámara si es necesario. Además, que ayuda en el trabajo de la animación debido a que muestra la secuencia en la cual todos objetos se deben comportar y que es lo deben hacer.

Tabla 12 .Storyboard de escena introductoria

Storyboard		
Nombre del proyecto: Actividad didáctica para el estudio del movimiento unidimensional		Actividad: Bienvenida e introducción
Descripción: Bienvenida de Galileo Galilei al usuario		Página: 1 – 2
	Diálogo	Traducción
	Hey, hey, tu Parces non essere il postro per te	(Hey, hey, tu no pareces ser de este lugar)
Sacude el brazo		
	¿Non capisci che non parlo italiano? ¿Parle francais? ¿Non? ¡O!! ¿Así que hablas español?	¿No me entiendes no hablas italiano? ¿Hablas francés? ¿No?
Hace reverencia para presentarse		
	¿Hola soy Galileo no pareces ser de por aquí dime que haces? Ummmmmm, A sí que también te preguntas que cae primero.	
Se toca la barbilla		

Storyboard	
Nombre del proyecto: Actividad didáctica para el estudio del movimiento unidimensional	Actividad: Bienvenida
Descripción: Bienvenida de Galileo Galilei al usuario	Página: 2 – 2
	<p>¡Pero yo el gran Galileo!!! Junto contigo claro le daremos respuesta.</p>
Señala al usuario	
	<p>Claro que en mi tiempo hay muchas limitaciones, así que debo arreglar unas tantas cosas antes.</p>
Señala el piso y actúa con prisa	
	<p>Anda regresa a tu tiempo y utiliza esos raros azulejos para comprobarlo. ¡Nos veremos Amico!!!!.</p>
Hace una señal de victoria	

Nota: La imagen del personaje utilizadas en el presente story board fueron tomadas del personaje 3D a través de Daz3D, el cual fue adquirido para hacer incorporado durante el desarrollo de la aplicación.

3.2.4 Descripción de componentes

En este apartado se describieron las características de los objetos (tabla 13) y espacios arquitectónicos (tabla 14) con el objetivo de definir y tener claro su aspecto. En el caso de los audios (diálogos), se realizó una lista de etiquetas con el objetivo de que cuando se desarrollen se puedan identificar para integrarlos de manera adecuada en las animaciones (tabla 15).

Tabla 13. Descripción de objetos

Identificador:	Tipo de plantilla:	Descripción de objetos:				
	Objeto:	Características:			Tipo de objeto:	Escenario:
		Color:	Forma:	Textura:		
Balón_B	Balón de básquet	Naranja y negro	Esférico	Cuero	Interacción	Luna, Júpiter, sima de la torre, cámara de vacío y marte
Bola_B	Bola de boliche	Negro	Esférico	Resina		
Portal_L	Portal luna	Negro y blanco	Circular	Paisaje		
Portal_M	Portal marte	Negro y rojo	Circular	Paisaje		
Portal_J	Portal Júpiter	Negro y amarillo	Circular	Paisaje		
Portal_CV	Portal Cámara de Vacío	Negro y gris	Circular	Paisaje		
Portal_MT	Portal mirador de torre	Azul y verde	Circular	Paisaje		
Lámpara	Lámpara	Gris y amarillo	Cubica	Metal	Contexto	Cámara de vacío
Bloqueo	Valla	Gris	Tubular	Piedra		Plaza de pisa

Tabla 14. Descripción de espacios arquitectónicos

Id:	Tipo de plantilla:	Descripción de espacios arquitectónicos:				
	Objeto:	Características:			Tipo de objeto:	Escenario:
		Color:	Forma:	Textura:		
Torre 1	Torre de pisa	Blanco	Tubular	Mármol	Contexto	Plaza de pisa
Domo	Duodomo	Rosa y blanco	Tubular	Mármol		
Catedral	Catedral de pisa	Rosa gris y blanco	Iglesia	Mármol y granito		
Mirador	Mirador de torre de pisa	Blanco y gris	Tubular	Mármol		
Torre 2	Torre de pisa	Blanco	Tubular	Mármol		Luna, Júpiter, sima de la torre, cámara de vacío, marte y plaza de pisa.

Tabla 15. Identificación de secuencias de audio

Tipo de plantilla:	Identificación de secuencias de audio	
ID Dialogo	ID Audios	Personaje
DI1, DI2, DI3	G11	Galileo Galilei
DI4	G21	
DI5	G31	
DI6	G51	
DI7, DI8, DI9, DI10	Todas2	
DC1	GC1	
DC2	GC2	
DJ1	GJ1	
DJ2	GJ2	
DL1	GL1	
DL2	GL2	
DL3	GL3	
DM1	GM1	
DM2	GM2	
DT1	GT1	
DT2	GT2	

3.2.5 Modelado de eventos

En esta actividad se desarrolló el modelo de eventos para describir los movimientos que podrá realizar el usuario través del dispositivo Leap Motion cuando esté se encuentre navegando dentro de los escenarios (Tabla 16) y las acciones que desencadena durante la experimentación de la aplicación (Tabla 17 y 18).

Tabla 16. Secuencia de eventos de desplazamientos

Tipo de plantilla:	Descripción de eventos de desplazamiento del usuario a través de leap Motion	
Eventos	Acción	
Introducir y mover mano izquierda.	Girar cámara sobre el X y Y.	
Introducir mano derecha.	Desplazamiento del personaje hacia delante.	
Mover mano derecha hacia la izquierda.	Desplazamiento del personaje y rotación de cámara hacia la izquierda.	
Mover mano derecha hacia la derecha.	Desplazamiento del personaje y rotación de cámara hacia la derecha.	

Tabla 17. Secuencia de eventos para dispositivo Leap Motion durante la experimentación

Tipo de plantilla:	Descripción de eventos de desplazamiento del usuario a través de leap Motion durante la experimentación	
Evento detectado	Acción	
Introducción de manos.	Creación de manos de entorno y sincronización.	
Contacto con objetos en entorno.	Colisión y desplazamiento de objeto.	
Acercarse a un objeto y tomar con pulgar e índice	Toma magnética del objeto.	
Contacto con botón.	Secuencia de botón.	

De la misma forma en la tabla 18, se describen los eventos de los objetos tridimensionales que tendrán interacción con el usuario u otros objetos.

Tabla 18. Secuencia de eventos con objetos

Tipo de plantilla		Descripción de eventos detectados por los objetos tridimensionales generados por el usuario .	
Componente Tridimensionales	Eventos detectados	Condiciones	Acción
Inicializador de Tutorial	Introducción de manos	Colisión con instanciado	Activación de secuencia de tutorial.
		Comprobación de avance	Comprobar tutorial.
Portal	Colisión con el personaje	Siempre se cumple	Cargar nuevo escenario.
Inicializador de experimentación	Colisión con el personaje	Si el usuario entra	Cambia de cámara principal a experimentación, e inicia Leap Motion.
Finalizador de experimentación	Colisión con la mano del personaje.	El usuario activa el botón en el escenario	Cambia de cámara del personaje y desactiva la cámara de experimentación, y el Leap Motion pasa su manejo correcto.
Instanciador de objetos	Colisión con el objeto al caer	Al pasar por el colisionado y el objeto es único en el mapa	Crea otro objeto de las mismas características destruyendo el anterior.
Cambio de cámaras	Colisión con objeto al caer	Si el objeto entra en contacto.	Se desactiva la cámara principal y se pasa a la cámara "Split" para ver la caída en varios ángulos.
Destructor	Colisión con cualquier objeto.	El objeto entra al coleccionador.	Cualquier objeto que entre al coleccionador será destruido y automáticamente, dará el pase para la creación de otro
Botón de decisión Galileo	Colisión con el personaje	El usuario toca el botón.	Activa la máquina de estado de galileo, correspondiente a la acción del usuario.
Área de interacción con galileo	Colisión con el personaje	El usuario entra al coleccionador	Galileo despliega la secuencia de animación y botones para su interacción con el
Área de animación de Galileo	El usuario hace contacto con el colisionador.	Si el usuario entra al coleccionador.	Galileo despliega la secuencia de animación.
		Si el usuario sale del coleccionador	Galileo detiene la secuencia de animación y regresa a la animación default.
Área de secuencia de Galileo	El usuario hace contacto con coleccionador.	Si el usuario entra al colisionador	Galileo despliega secuencia de animación.
		Si el usuario sale del colisionador	Galileo inicia una secuencia de tiempo para después activar la siguiente animación del escenario.

3.3 Fase de Desarrollo

En esta fase se desarrollaron los recursos artísticos, buscando siempre ajustarse a todos los requerimientos obtenidos en la fase de análisis y bajo el siguiente orden:

- Se modelaron y optimizaron los objetos tridimensionales.
- Se texturizaron los objetos tridimensionales.
- Se animaron los objetos tridimensionales que lo requerían.
- Se integraron los objetos tridimensionales en el motor de representación.
- Se programaron los scripts necesarios para el funcionamiento de la aplicación.

3.3.1 Producción de recursos artísticos

3.3.1.1 Modelado

Debido a que es un entorno virtual que trata de simular el mundo real, se realizaron modelados tridimensionales en escala 1:1, esto es que, se necesitaron obtener las medidas reales de los objetos a representar. Para la creación de los diferentes monumentos que se utilizan en el presente proyecto, se necesitó consultar literatura pertinente para poder obtener las dimensiones de cada uno de ellos.

El software que se ocupó para realizar los recursos artísticos tridimensionales, son: Autodesk Maya y DAZ Studio 3D

Las técnicas de modelado tridimensional utilizadas para este proyecto son varias, por ejemplo: Box-Modeling, Geométrico, por texturizado y booleano

Para la creación de los diferentes monumentos se ocupó el software Autodesk Maya , se inició utilizando figuras base (Box modeling) que se asemejaran a la estructura final. Cabe señalar que se debe tener en cuenta la medida de estos elementos ya que determinara la medida final del componente (en este caso la torre de Pisa), ver figura 30.

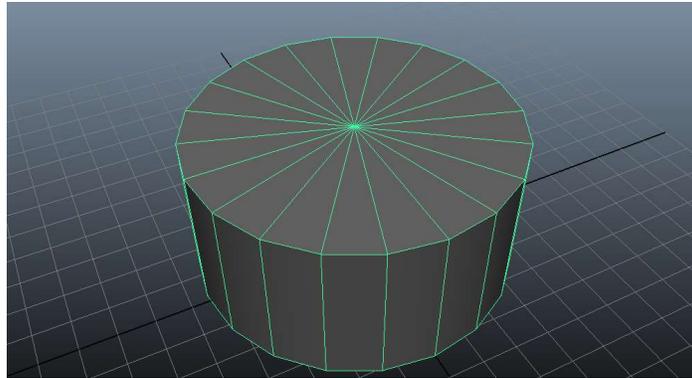


Figura 30. Base torre

Al tener la estructura base se agregaron ejes (Modelado Geométrico) los cuales sirven para realizar los cortes necesarios para crear los distintos arcos que conformaran la torre, como se puede ver en figura 31.

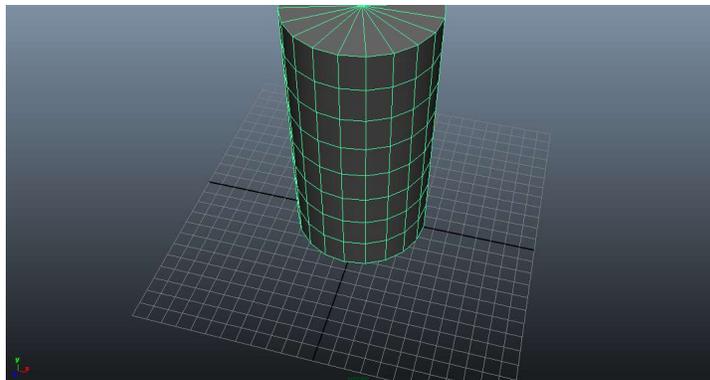


Figura 31. Cortes torre

A continuación, para crear los diferentes pisos de la torre se utiliza la herramienta “Extrude face”(figura 32) que proyectara las caras de los ejes que agregados de esta manera se simulan los pisos de la torre.

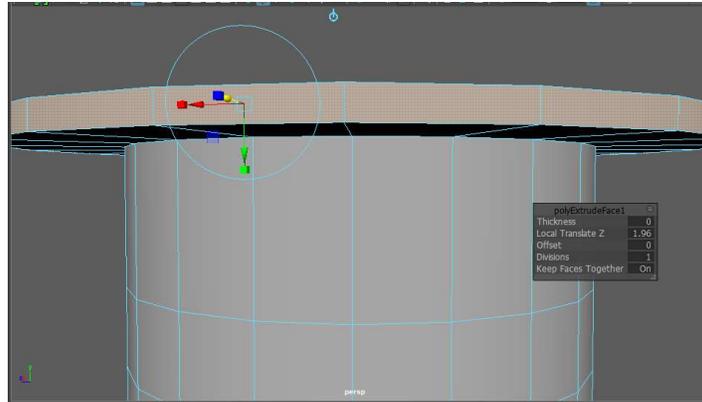


Figura 32. Pisos de la torre

Posteriormente, se agregan figuras (modelado booleano) con forma de herradura, los cuales sirvieron de guía para realizar los cortes de los arcos en cada piso como se muestra en la figura 33.

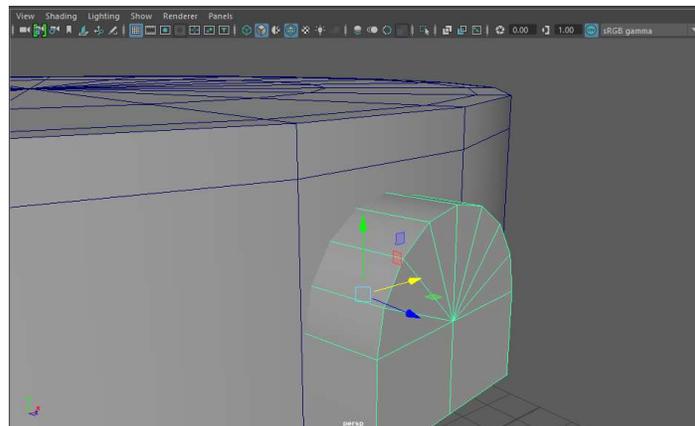


Figura 33. Arcos de la torre

Una vez logrado lo anterior se revisa que los cortes (modelado booleano) sean los correctos ya que estos pueden causar problemas durante la optimización agregando o generando caras escondidas (Overlapping), como se observa en la figura 34.

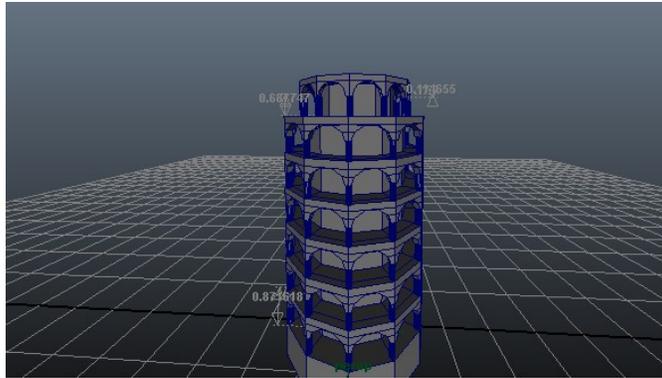


Figura 34. Arcos correctos de Pisa

3.3.1.2 Optimización

Al tener los cortes creados se inicia con la optimización del modelo, mediante la técnica N-Gons, esta se logra buscando de manera automática caras con más de 4 lados y agregando ejes a las caras triángulos. Los cuales ayudan a *Unity* en la representación en tiempo real, ya que esto no tiene que ser calculado por el motor ejemplo de la figura 35.

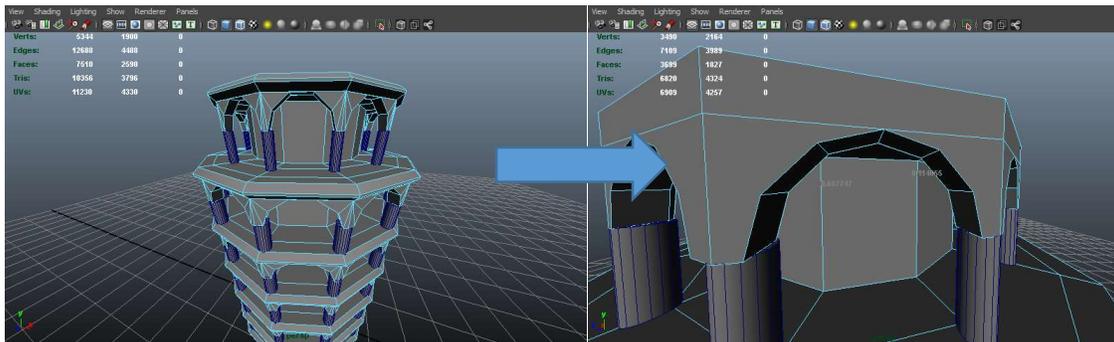


Figura 35. Agregado de ejes

Al terminar con el modelo 3D se inicia con la texturización, la cual consiste en darle color a nuestro objeto para lo cual se crea un UV map en Maya, que es el desarmado de nuestra figura 3D en un plano 2D (figura 36).

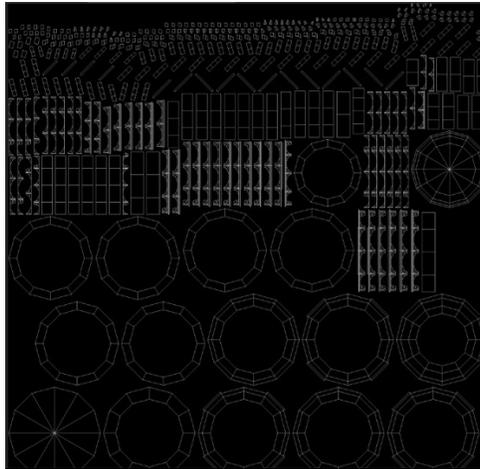


Figura 36. Uv map torre de pisa.

Al tratarse de monumentos se intenta tener la mayor recreación de este, por lo cual se utilizaron imágenes y fotos obtenidas en la fase de diseño, las cuales son acomodados de manera muy específica en cada sección, observar figura 37.

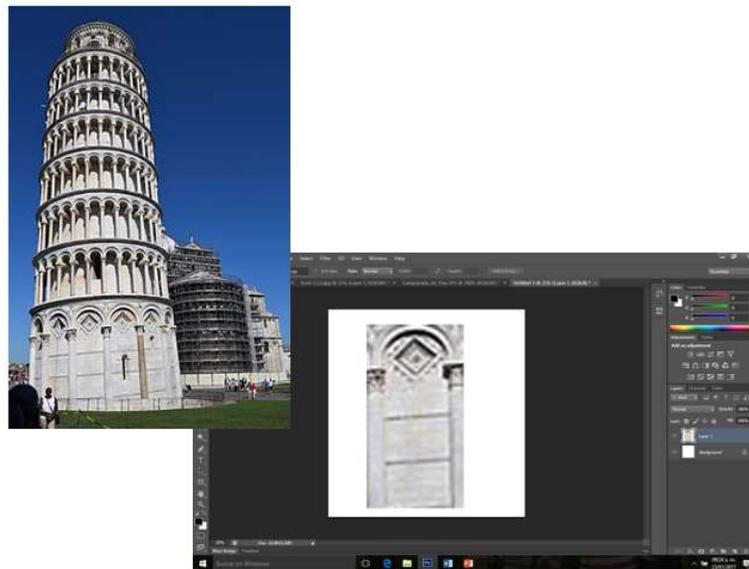


Figura 37. Agregado de imagen

Cada una de estas secciones debe ser cortadas y colocadas en el punto exacto de los UV para que al armar estas encajen de manera correcta, eliminando con esto que se generen marcos en las formas (figura 38).

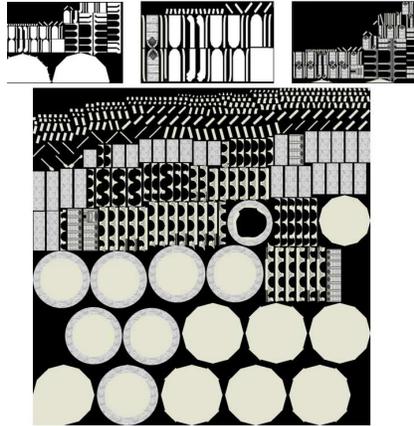


Figura 38. Corte de imágenes

Una vez que se realiza el procedimiento anterior se pasa a *Autodesk Maya* para el armado y a través de la herramienta UV texture, y se observa que, en todas las caras se tengan los rasgos deseados del y que sean acorde al monumento real como se ve en la figura 39.

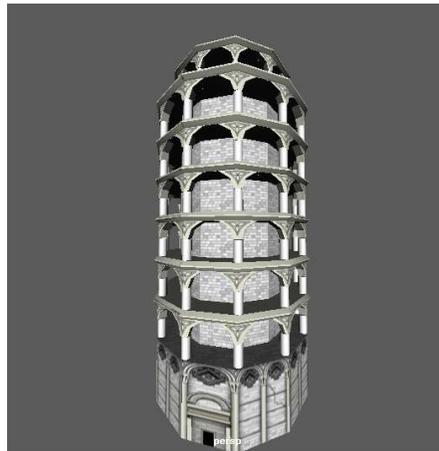


Figura 39. Modelo texturizado

3.3.2 Producción de audio

Para realizar el proceso de captura de secuencias de audio, que se convertirán en la voz de Galileo Galilei, se utilizó un micrófono USB, la grabadora de voz de Windows (Figura 40), y un cuarto solo sin mucho eco para lograr una grabación lo más limpia posible.

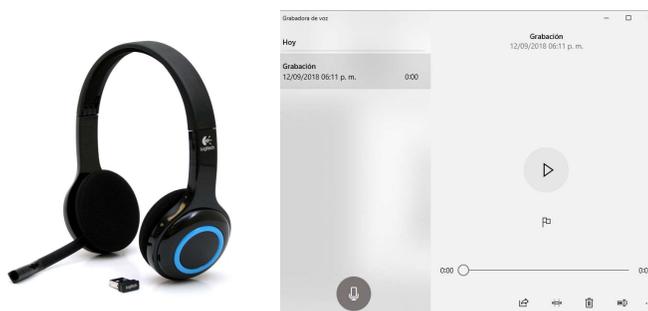


Figura 40. Dispositivos de grabación

Una vez que se grabaron las secuencias se empezó el proceso de limpieza de las mismas, mediante el uso del programa Audition de Adobe, se analizó las secuencias de audio, en búsqueda de eco o ruido en la grabación para que no interfiriera con la correcta reproducción de la voz (Figura 41).



Figura 41. Vista de ondas de sonido

Se pasa la grabación mediante la vista de onda o espectral (Figura 42), este tipo de visualización nos ayuda a poder ver las zonas con ruido o eco dentro de la grabación y separarlo de las partes necesarias.

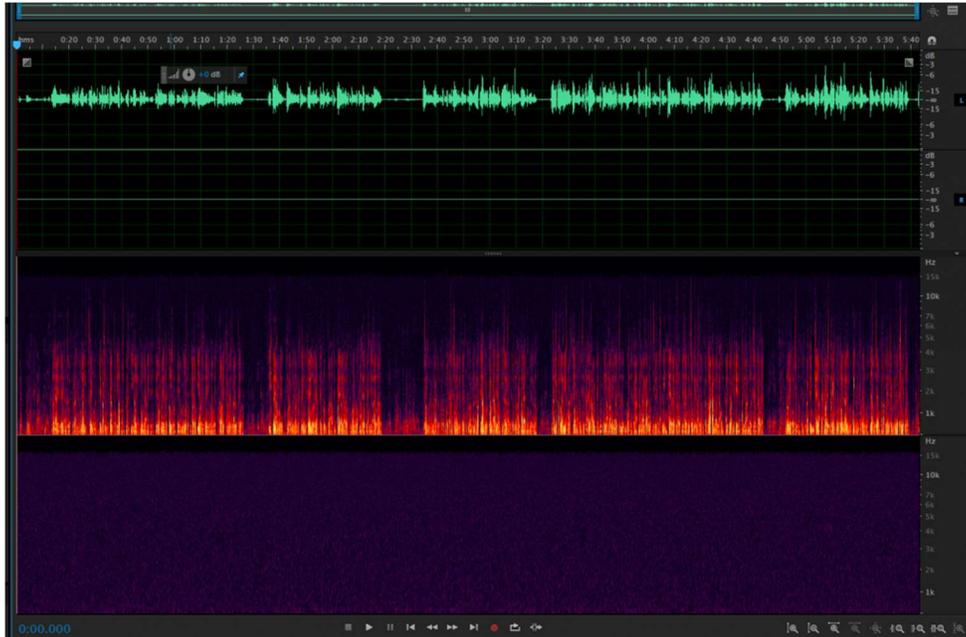


Figura 42. Vista de espectro

Con ayuda del efecto de “reducción adaptativa de ruido” que cuenta Adition, de manera gradual se fue reduciendo los sonidos y ecos que no eran deseados en la grabación (figura 43).

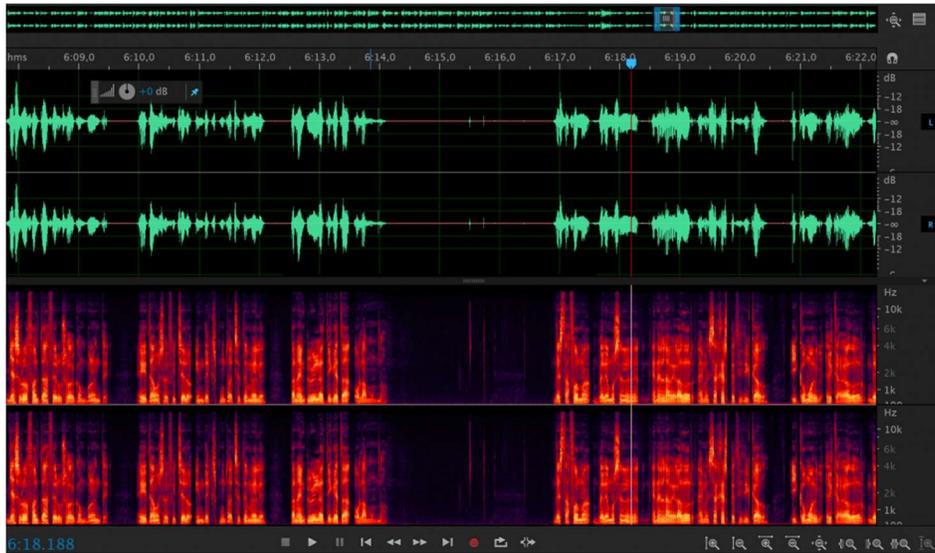


Figura 43. Vista normalizada

3.3.3 Creación y optimización de personajes de Daz3D

Una vez que se terminaron todas las estructuras y los objetos 3D se editó el personaje principal con ayuda del programa Daz 3D. Este programa nos permite obtener personajes prefabricados y personalizables que contienen una estructura ósea, texturas, y paquetes de animación, ahorrándonos mucho tiempo de desarrollo.

Dentro de Daz 3D se inicia seleccionando una anatomía en la interface de “Actors, Wardrobe, & Props”, la cual sirvió como base del personaje. En este caso se utilizó “Genesis 2 Base Male” (figura 44) para desarrollar a Galileo Galilei.

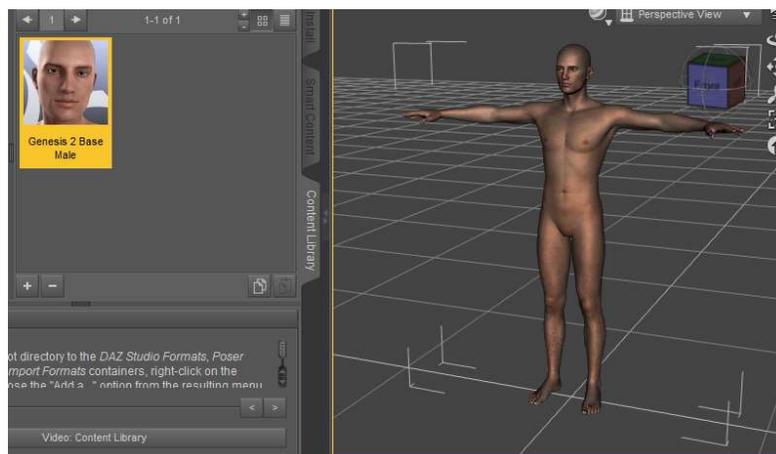


Figura 44. Anatomía en Daz 3D

Después de esto se continuo con la vestidura y caracterización del personaje (figura 45) Para ello se seleccionó ropa y cabello dentro de la opción “Actors, Wardrobe, & Props”, que se encuentran incluidos en los assets de Daz de manera gratuita o con costo, los cuales se pueden adquirir mediante su plataforma. Cabe señalar que se puede encontrar diversas vestimentas, materiales para la piel, ojos, pelo, accesorios e incluso animaciones.



Figura 45. Selección en Daz 3D

Una vez que se editó el personaje con las características deseadas, fue necesario agregar el “Face Key”. Está es una animación dividida en diferentes expresiones faciales que es utilizado, para para crear tanto la animación corporal como la facial y tener toda compatibilidad con Iclone 6 de real illusion. (figura 46).

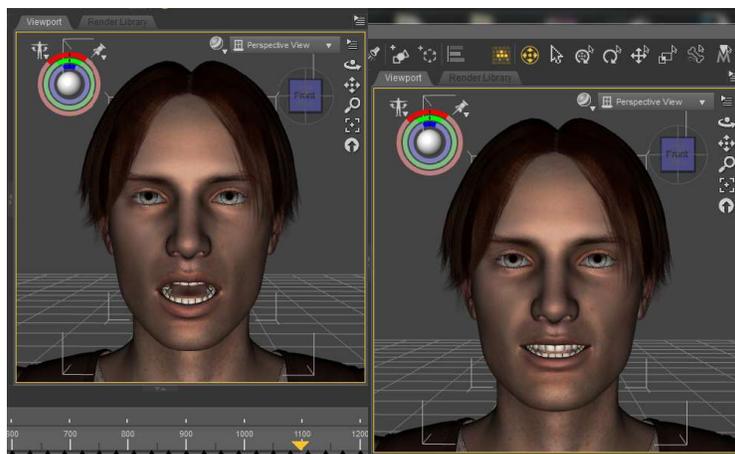


Figura 46. Facekey en Daz 3D

Posteriormente, el personaje fue exportado en formato fbx. para ser animado tanto corporal como facialmente en Iclone. Sin embargo, el tamaño del personaje no fue el más óptimo para trabajar en entornos de tiempo real, por lo cual fue necesario optimizarlo a través de Simplygon, que es una herramienta de optimización automática que permite reducir la cantidad de polígonos y tamaño de los materiales.

Para ellos, se inicia el programa simplygon y se arrastra el archivo en formato fbx. en la zona de trabaja para importarlo a la aplicación como se ve en la figura 47.

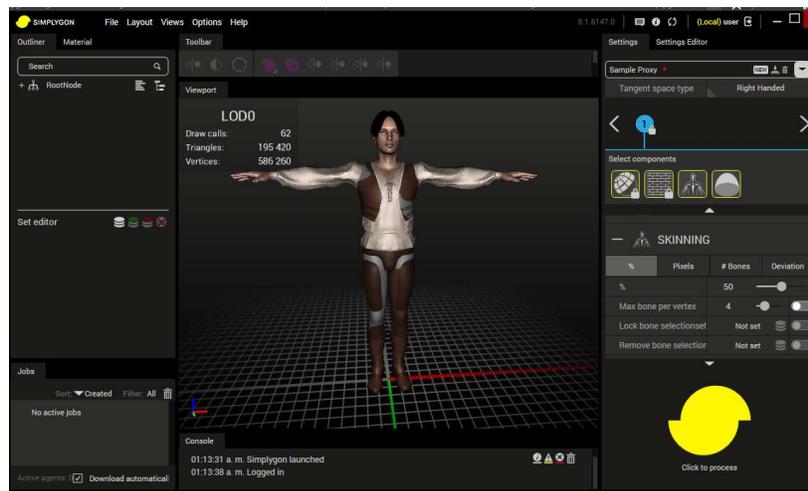


Figura 47. Personaje en simplygon

Dentro de la aplicación tendremos la opción de optimizar la estructura de modelado del personaje de manera general, reduciendo un porcentaje específico en todas sus partes o de manera específica, reduciendo los polígonos por sección. Esto ayuda a optimizar más aquellas secciones del personaje que no son indispensables o tienen poca relevancia dentro de la aplicación.

En el inspector de simplygon encontraremos todos aquellos modelos o secciones de maya que conforman el personaje, en este caso se toma de ejemplo la ropa (figura 48).

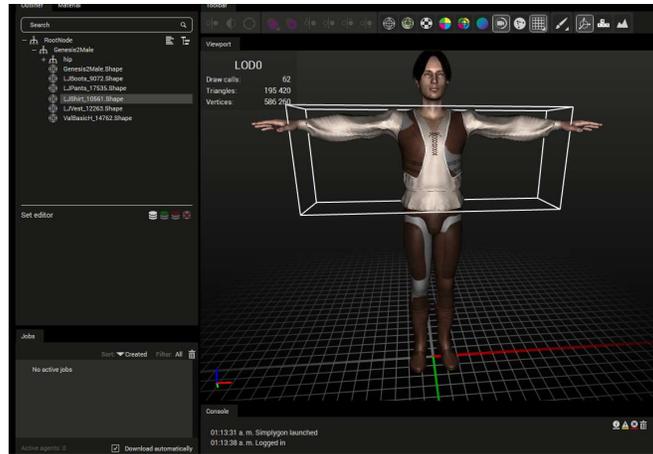


Figura 48. División del personaje en sympligon

Con dichas opciones se redujo la ropa un 40%, mientras la anatomía del personaje un 60%, debido a que se busco que las animaciones faciales no fueran alteradas por la optimización (Tabla 19).

Tabla 19. Comparativa de optimización

Comparativa de optimizacion		
Herramienta de optimizacion	Número de poligonos original	Polígonos después de la optimización
Manual	47426	39812
Polycrouncher		20481
Sympligon		17473

Cabe señalar que al reducir demasiado los poligonos el modelo puede sufrir errores al ejecutar las animaciones, por ejemplo el personaje se rompe o pierde naturalidad al ejecutar la animación que le permite caminar dentro de un escenario (figura 49).



Figura 49. Vista comparativa de optimización

Simplygon ofrece una gran cantidad de vistas comparativa (figura 50) para poder evaluar el personaje antes de su exportación.

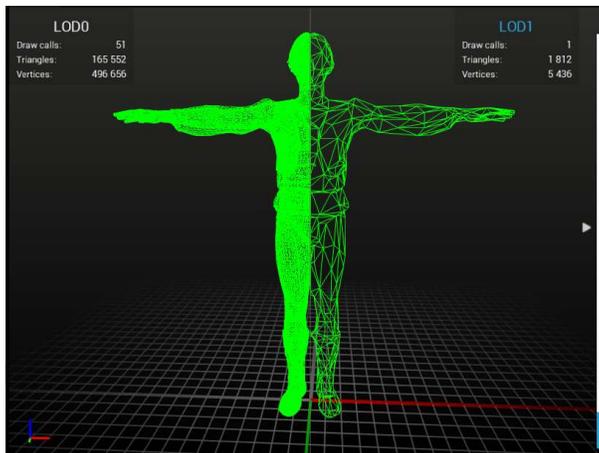


Figura 50. Vista comparativa de la maya

Una vez que se tiene el modelo optimizado se exporta en formato fbx., donde simplygon genera los materiales y modelo.

3.3.4 Animación

Después de la creación del personaje en *Daz3D* y su optimización en *sympligon* se inició con el proceso de animación corporal y facial del personaje utilizando la técnica de animación ortoscopia (Captura de Movimiento), que ayuda a facilitar el proceso de animación, copiando los movimientos de un actor, sin la necesidad de hacerlo frame por frame con *3Dxchange* se siguieron los siguientes pasos:

1. Se inicia con el proceso de identificación de huesos, como es un personaje creado en *Daz 3D* los huesos del personaje se encuentran preclasificados como un bípedo humanoide, esto ayuda a *Iclone 6* a su identificación se puede ver en la figura 51.

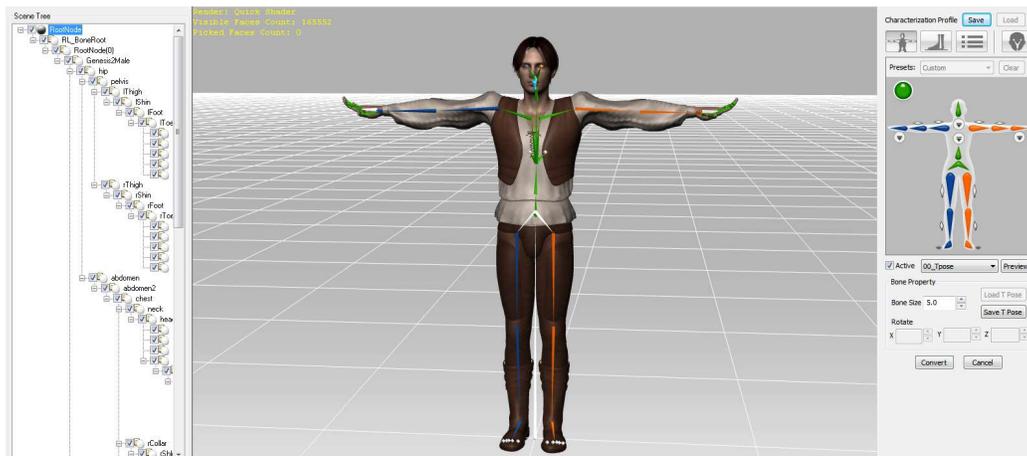


Figura 51. Integración de huesos

2. Se exporta a *Iclone* presionando “Apply to Iclone”. De manera automática se abrirá *Iclone* con nuestro personaje en el entorno de trabajo (figura 52).

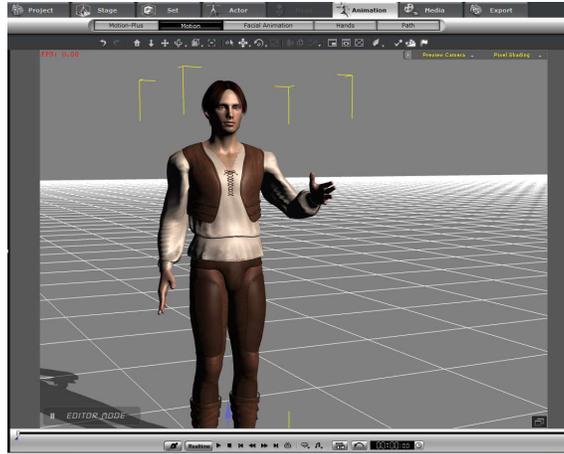


Figura 52. Inicio de animacion

3. Se realiza el proceso de animación, para este caso se utilizó la herramienta *Mo-cap* y *Kineckt* de *Xbox 360* de *Microsoft*, que permite la captura de movimiento corporal de un actor que posteriormente será reproducido por el personaje 3D cuadro por cuadro. Para lo cual, dentro de la aplicación *Mo-cap* se presiona el botón de conexión, este abrirá la ventana que puede verse en la figura 53, automáticamente inicia la búsqueda del dispositivo *Kineckt* de *Xbox 360* y establece el vínculo con *Mo-cap*.



Figura 53. Conexión con kineckt

A continuación, se calibra la detección de movimiento, siguiendo las siluetas dadas por *Mocap* como se muestra en la figura 54 y al terminar el personaje toma la posición del actor que realiza la captura de movimiento.

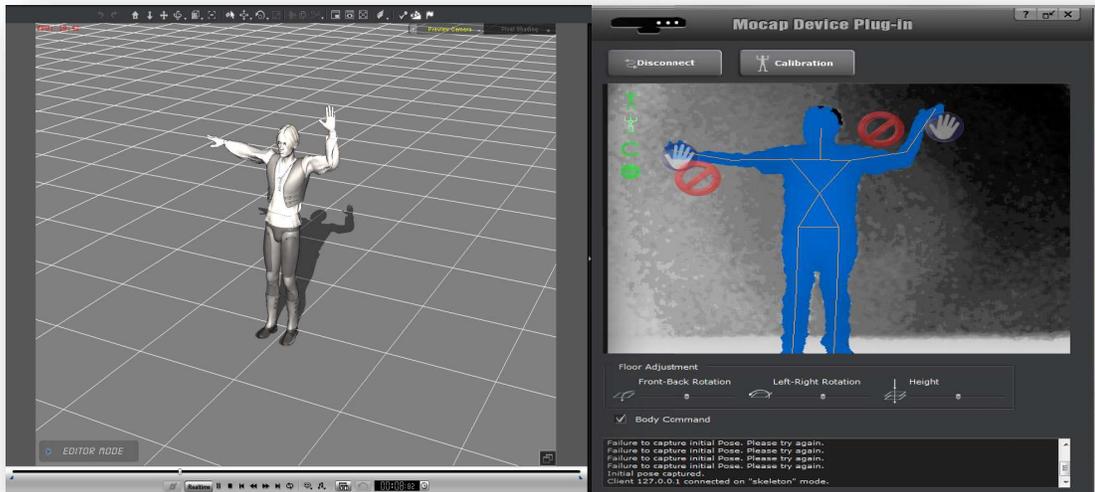


Figura 54. Sincronización kinect y iclone

Teniendo en cuenta las diferentes limitaciones de la detección del dispositivo *kinect* se evitaron los cruces de manos y se tuvieron que exagerar todos movimientos dentro el área de detección (figura 55), para lograr la animación deseada.



Figura 55. Secuencia de animación.

4. Posteriormente se utilizó el *Puppet Animator* de Iclone 6 para completar la animación, esta herramienta permite detallar el movimiento de las manos, el cuello, dedos y la cara ya que no son capturados por el *kinect*, (figura 56).

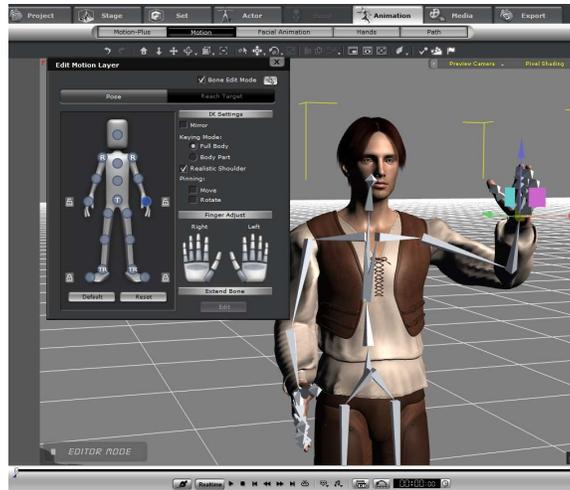


Figura 56. Edición de animación corporal

5. En esta misma aplicación se realizó la animación facial agregando el audio directamente en *Iclone* para que el programa generara automáticamente la animación y se pulió agregando expresiones y gestos en cada palabra del diálogo, dando como resultado una animación natural y fluida (figura 57).



Figura 57. Animación facial

6. Después se exporta el personaje, utilizando la opción “*Back to 3Dxchange*”, para que el personaje 3D regrese a *3Dxchange* junto con el clip de animación, como se muestra en la figura 58 .

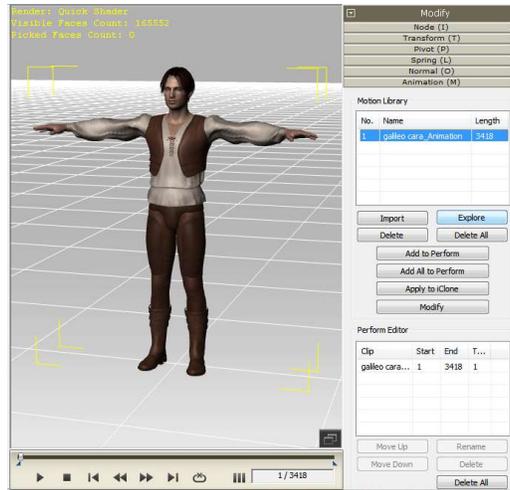


Figura 58. Integración de elementos en fbx

- Una vez integrada cada animación en el personaje, se exporta para su uso en *Unity*. *3Dxchange* tiene la opción de exportar directamente a *Unity*, pero normalmente esta exportación causa conflicto con animaciones faciales. Por lo cual se exporto, un segundo personaje en formato fbx el cual contiene todas las animaciones requeridas (figura 59).

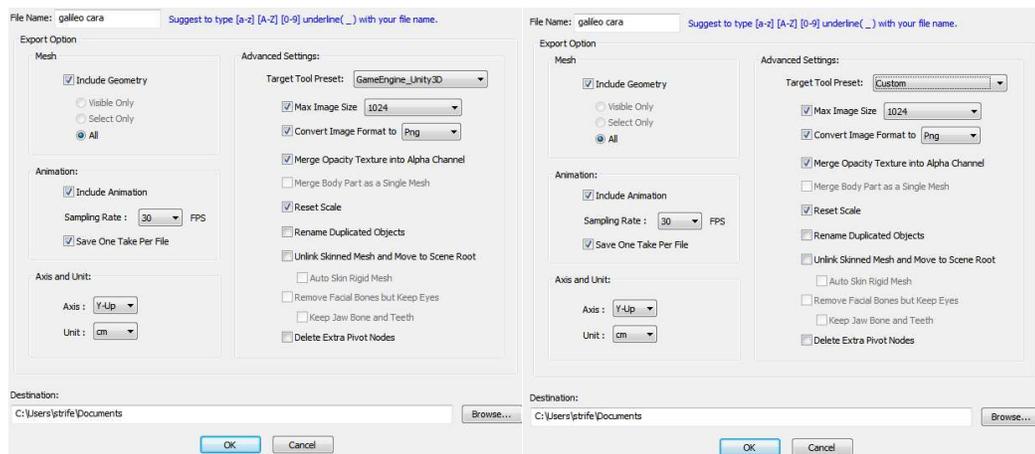


Figura 59. Exportación de fbx

3.3.5 Skybox

Un skybox es la bóveda celeste o cielo que se representa en el escenario dentro de los entornos de RV.

Para obtener un mayor nivel de inmersión fue necesario crear diferentes Skybox 's para que fueran lo más cercano posible a la bóveda celeste de cada planeta como se puede observar en el ejemplo de la figura 60.



Figura 60. Skybox lunar

A continuación se explican como crear un skybox a partir de un conjunto de imágenes:

1. Dentro del motor de representación, se creó un material especificando que es un *Skybox* de seis secciones y se agregaron las imágenes según su posición (figura 61).

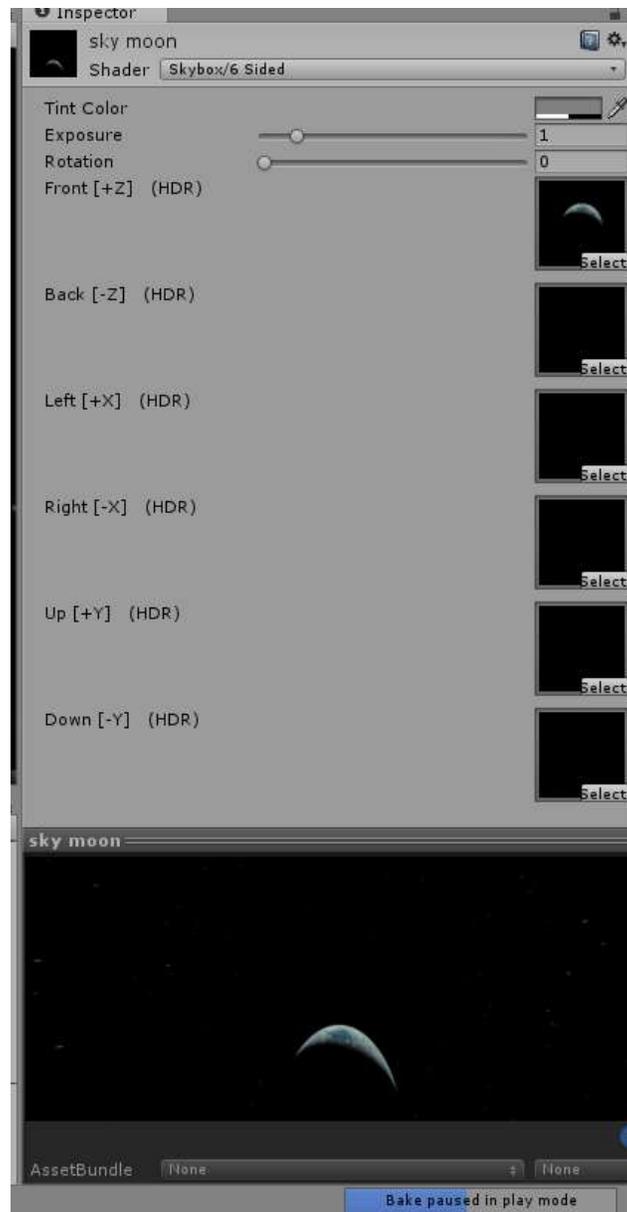


Figura 61. Armado de skybox en unity

2. Cuando el skybox no es compatible generan marcos que rompen con la ilusión de una bóveda celeste real, por lo cual se tienen que tratar en *photoshop*, como se ve en la figura 62.



Figura 62. Borrado de margen

Sin embargo, en algunas ocasiones el skybox sigue sin coincidir de manera perfecta por lo cual se crean cube maps. Que consiste en hacer una sola imagen (cabe destacar que esta opción solo puede ser utilizada en *Unity 5* y versiones posteriores) como se explica a continuación:

1. Es necesario tener una plantilla de *Cube map* compatible con *Unity* como la siguiente (figura 63).

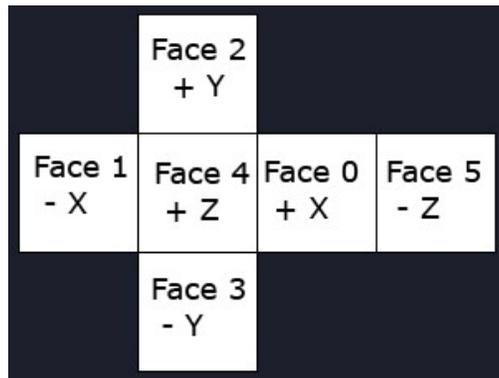


Figura 63. Mapa de cubemap

2. Para que el cubemap pueda ser utilizado como guía y ser rellenada con la imagen de skybox se agrega a *Photoshop CS5* (figura 64), para hacer los ajustes correspondientes.

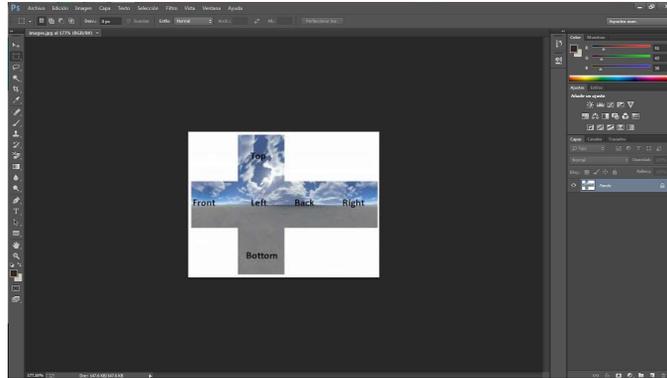


Figura 64. Creacion de textura

3. Debido a que plantillas se encuentran en una muy baja resolución se generaron dos texturas la primera sirve de referencia y la otra que será el *Skybox* final.
4. Dentro de Photoshop se crea un proyecto, en el cual se calcula con exactitud el tamaño de cada imagen que compone el *Skybox*, para evitar la pérdida de calidad en las imágenes (figura 65).

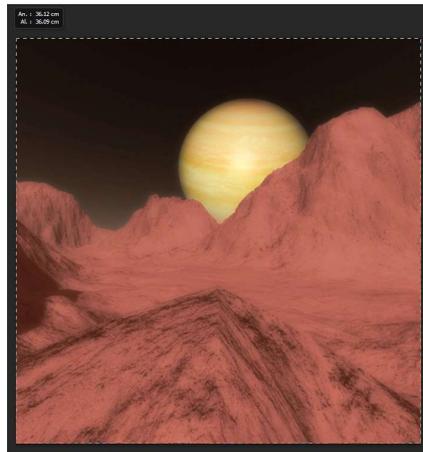


Figura 65. Toma de imagen de skybox

En este caso se utilizó una figura que mide 36.12 pixeles de ancho y 36.9 de alto y una plantilla de tres por cuatro, por lo cual la imagen final mide tres veces de largo y cuatro veces de ancho mas que la imagen original, como se puede observar en la figura 66.

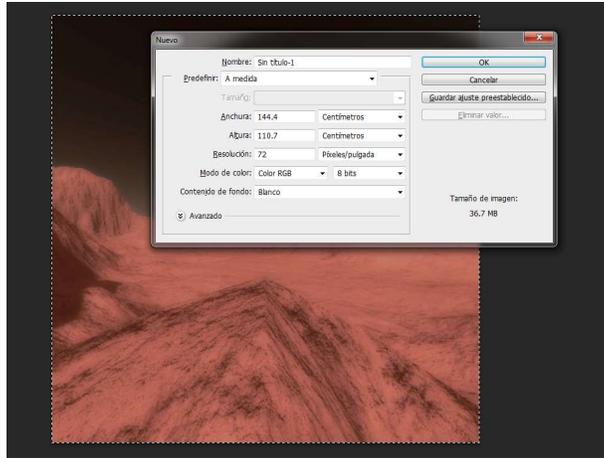


Figura 66. Ordenado de imágenes

5. Se juntó la plantilla que se utiliza como guía para que se rellene con las imágenes (figura 67).

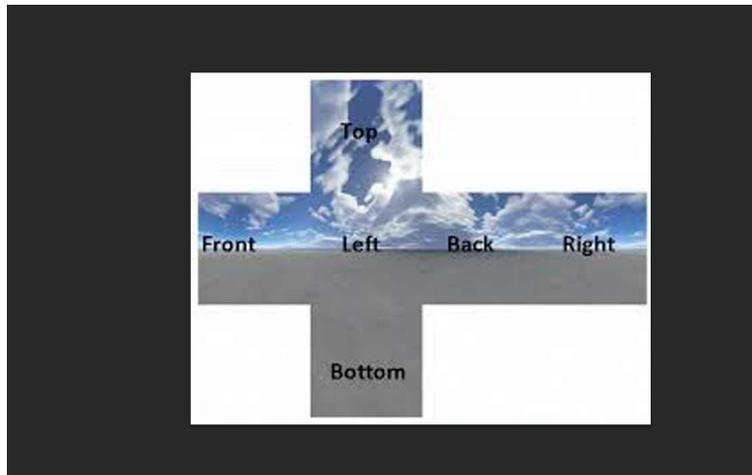


Figura 67. Plantilla en photoshop

6. Después se ajusta la plantilla a las medidas y definición correcta, se rellena con imágenes de manera exacta tomando en cuenta el orden y la dirección de la plantilla, cuidando en especial los bordes ya que estos deberán pasar de manera imperceptibles en el cube map. Cabe señalar que se debe tener especial cuidado que ninguna de las imágenes superará el margen de la plantilla original debido a que una vez que unity detecte y arme el cubemap no será capaz de diferenciar entre puntos vacíos de la imagen y la imagen que se deseó armar (figura 68).

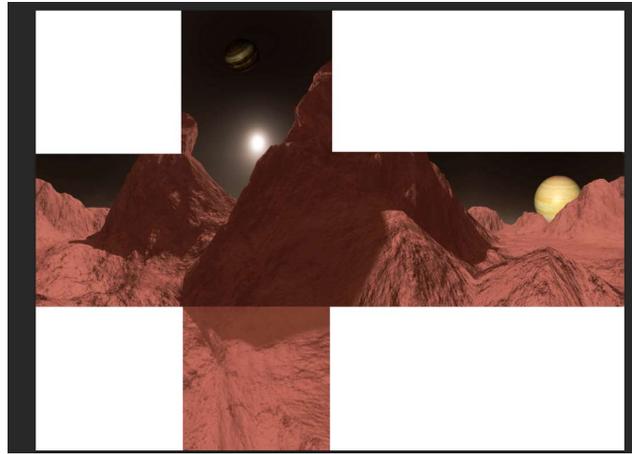


Figura 68. Cubemap finalizado

7. El *cube map* fue guardado en formato *targa* para que automáticamente lo reconozca *unity* y activando la opción “*Cube map*” para que lo arme de manera correcta (figura 69).



Figura 69. Cubemap en Unity

- Finalmente armado se buscó presencia de puntos o sectores en blancos ya que estos representan errores de armado. Una vez confirmado el armado de la textura se creó un objeto vacío esté especificado como un *Skybox* con textura de *cubemap* (figura 70) a diferencia de los *Skybox* anteriores este no es ensamblado por seis imágenes.

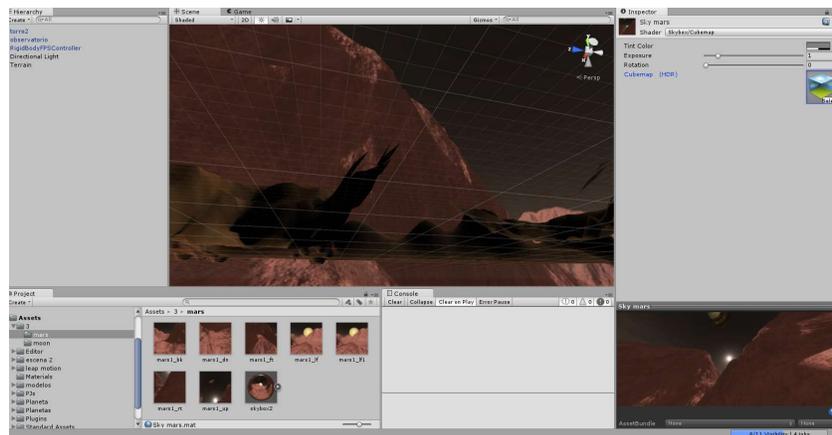


Figura 70. Cubemap en ejecución

3.3.6 Integración

En esta fase, se crearon los escenarios 3D para integrar los diferentes componentes desarrollados (objetos, estructuras, personajes y animaciones) con el objetivo de que el usuario navegó a través de ellos y experimenté el fenómeno de caída libre.

Tras haber obtenido todos los elementos que conforman la aplicación se crearan los escenarios de la siguiente manera:

- Se creó un proyecto nuevo, para unir todos los elementos de esta manera crear la aplicación final (figura 71).

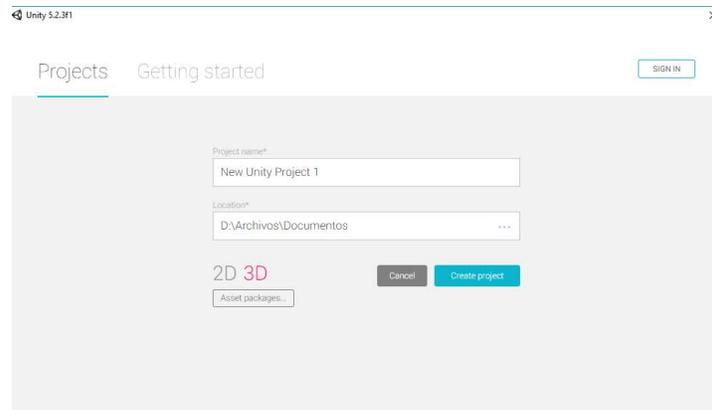


Figura 71. Proyecto unity

2. Dentro de *unity* se creó una escena y un terreno, en él se agregaron diversas texturas para hacer la reproducción de la plaza de pisa (figura 72).

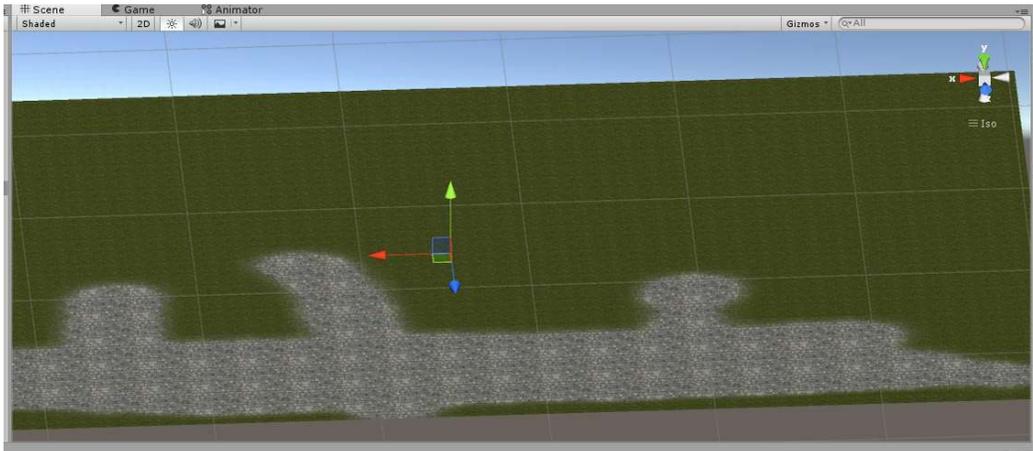


Figura 72. Terreno

3. Se exportaron uno a uno todos los objetos, cuidando que los modelos 3D se encuentre correctamente sin errores en su estructura, al igual que todas sus texturas. Estos elementos aparecen automáticamente en el inspector de *unity*, si son correctamente integrados (figura 73).

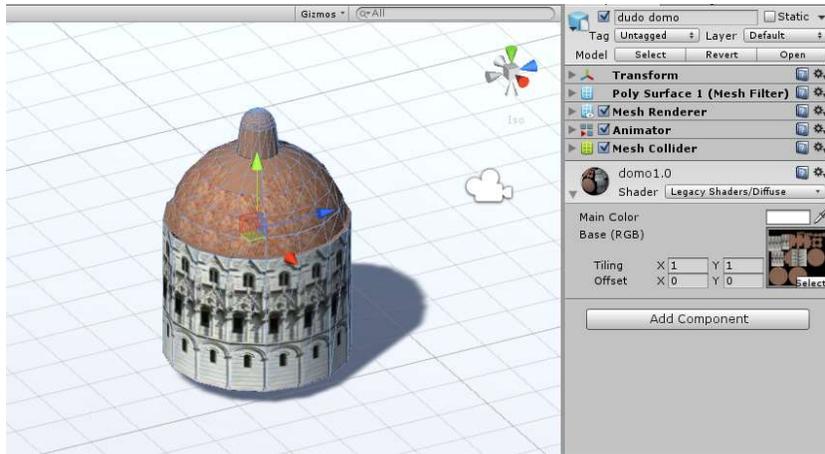


Figura 73. Estructura exportada a Unity

4. Se agregaron colisionadores y el atributo “rigibody” para que el objeto, sea afectado por la física del escenario (gravedad, masa, e interaccion con otros objetos). Una vez que el objeto tiene todos los atributos necesarios, se coloco dentro del escenario.
5. Los objetos fueron colocados según los diseños de los escenarios creados en la fase de diseño (figura 74).



Figura 74. Escenarios en Unity

6. Después, se exportaron los personajes y animaciones. Los cuales se encuentra conformado por una maya y texturas, a diferencia de los modelo estáticos, además estos cuenta con pequeños clips que representan las animaciones (figura 75).

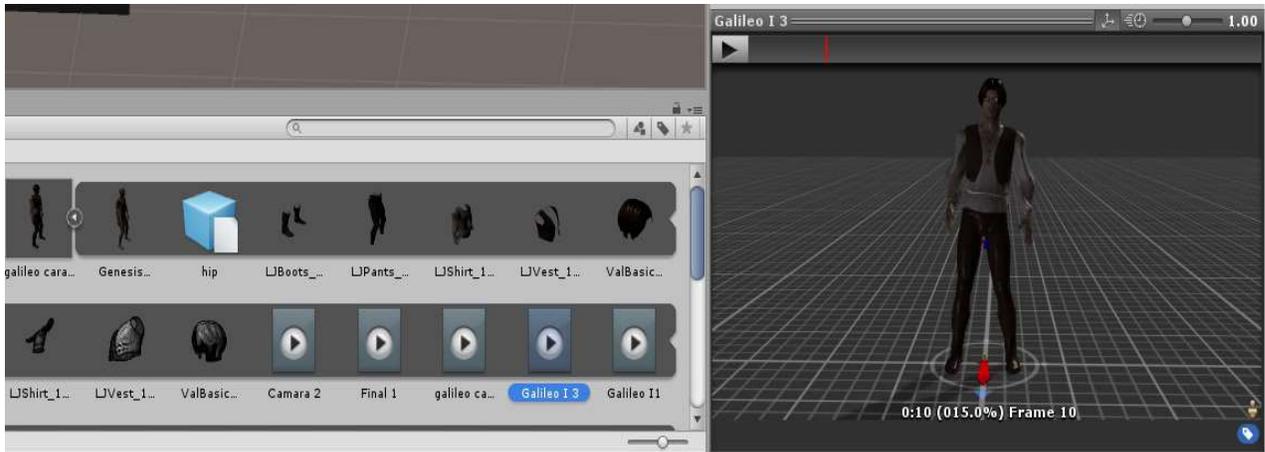


Figura 75. Personaje en Unity

En este caso al venir de un programa de desarrollo como *Daz 3D* y pasar por *Iclone*, algunos materiales y características fueron perdidas o no son totalmente compatibles, por lo se tienen que corregir para lograr tener una representación correcta.

7. Dentro del inspector el primer material que se arreglo fue el de la *córnea* del ojo, especificando que la textura era una transparencia y dando nuevas coordenadas a los mapas para alinear los ojos, como se ve en la imagen (figura 76).

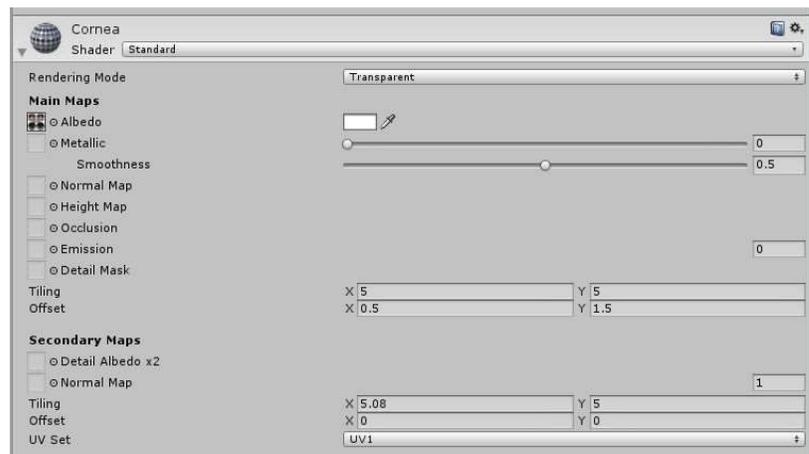


Figura 76. Material cornea

8. El segundo material a configurar fue el *EyeReflection*, al igual que el anterior se tuvo que recalibrar las coordenadas de los mapas y dar atributos de transparencia para corregir la representación de los ojos(figura 77).

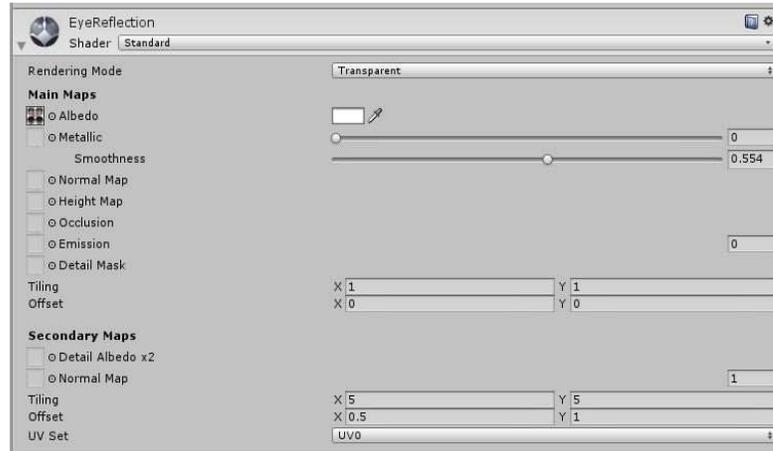


Figura 77. Material Eye reflection

9. A continuacion se modifico el material *M5PhillipLashes*, que contiene la textura de las pestañas (figura 78).

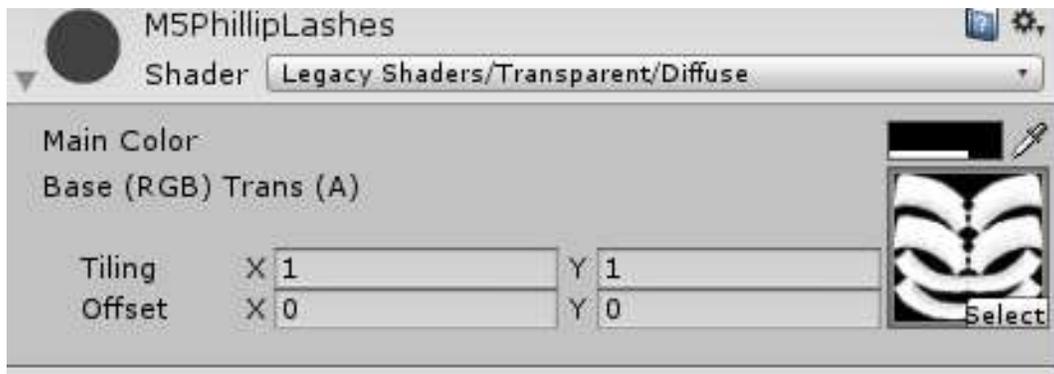


Figura 78. Material Pestañas

10. En este material en particular fue necesario recurrir a otra textura para su correcta representacion. Que se encuentra dentro del paquete de texturas del personaje, debido a que *unity* no es el unico en presentar este problema (figura 79).



Figura 79. Paquete de texturas Daz

11. Después de seleccionar esta imagen, debe especificarse que la textura cuenta con transparencias y debe habilitarse como una escala de grises (figura 80).

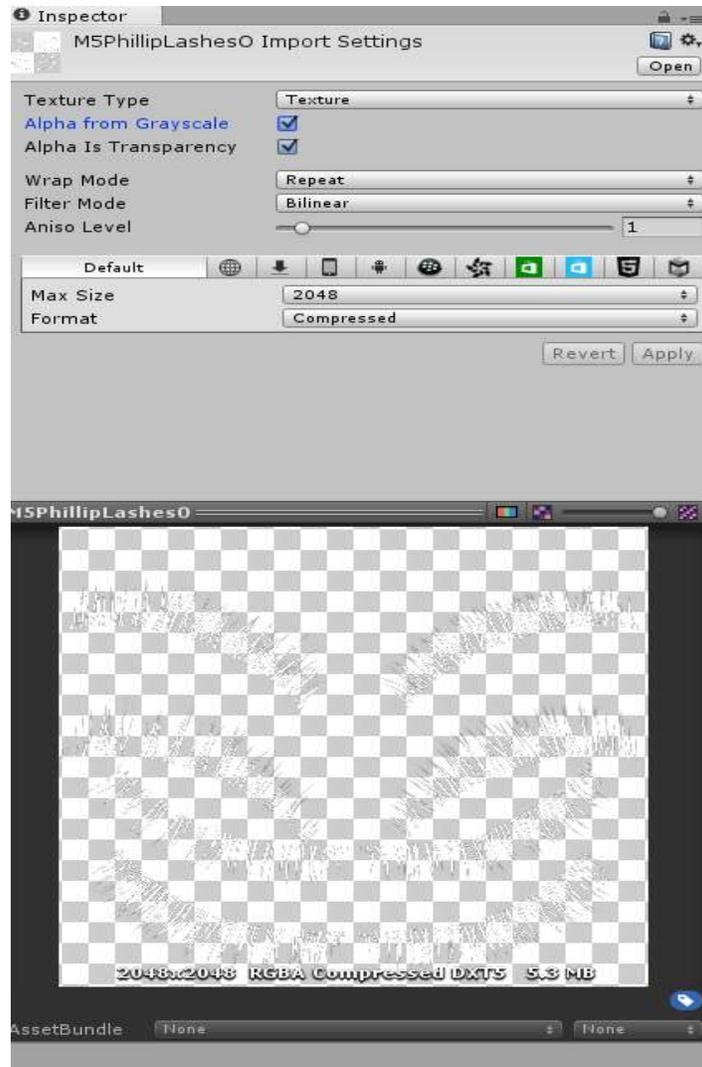


Figura 80. Transparencia en textura

12. Finalmente se agrega al material para concluir con las correcciones (figura 81).

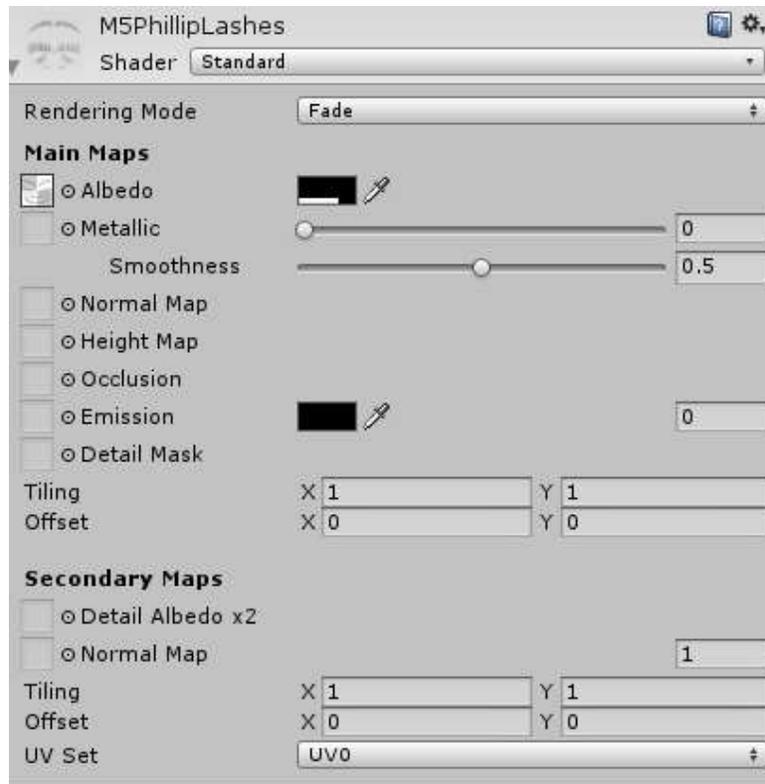


Figura 81. Material Pestañas final

3.3.7 Integración de lep motion a la aplicación.

Es un dispositivo capaz de capturar e interpretar el movimiento de los huesos de las manos para interactuar con la computadora. Puede ser utilizado directamente como un dispositivo de entrada (un mouse o teclado) o como dispositivo de RV.

Para poder utilizar el Leap Motion fue necesario importar el “Core Assets” en el proyecto de unity (disponibles en la página oficial de Leap Motion) compatible según la versión de Unity utilizada, en este caso se utilizó el Core Assets V 2.3.1 que es compatible con Unity 5.1 y posteriores.

Una vez que el “Core Assets” es descomprimido dentro de Unity, se encuentran 5 carpetas como se puede observar en la figura 82, que contienen todos los elementos necesarios para hacer uso Leap Motion.

A continuación se describen:

- **Gizmos:** Contiene una imagen de referencia de un Leap Motion.

- **OVR:** Contiene ejemplos y prefabs para el uso del dispositivo Oculus Rift.
- **Leap Motion:** Contiene ejemplos y prefabs para uso del Leap Motion en diferentes escenas, con o sin interacción.
- **Leap Motion + OVR:** Contiene ejemplos al igual que la anterior, pero estos fueron hechos para trabajar con el Oculus Rift.
- **Plugins:** Contiene el enlace de Unity con el Leap Motion.



Figura 82. Carpetas de “Core Assets”

Para fines del presente proyecto se utilizaron las carpetas: Leap Motion y Plugins.

Se debe ser cuidadoso al utilizar los diferentes prefabs, ya que esta decisión determina como el Leap Motion detectara las manos.

El prefab contenido en la carpeta *Leap Motion*, permite que el dispositivo este orientado en la detección desde el escritorio o desde abajo (figura 83). En el caso de que se elija el prefab contenido en *Leap Motion + OVR* se tendría que contar el adaptador Oculus y las manos serian detectadas desde arriba a la altura de la cara como se observa en la figura 84.



Figura 83. Detección de escritorio



Figura 84. Adaptador Oculus

En este proyecto se aplicó la opción Leap Motion junto con el script PluginLeapNotice. El cual tiene todas las opciones predeterminadas que permite hacer el enlace entre Unity y el dispositivo Leap Motion (figura 85).

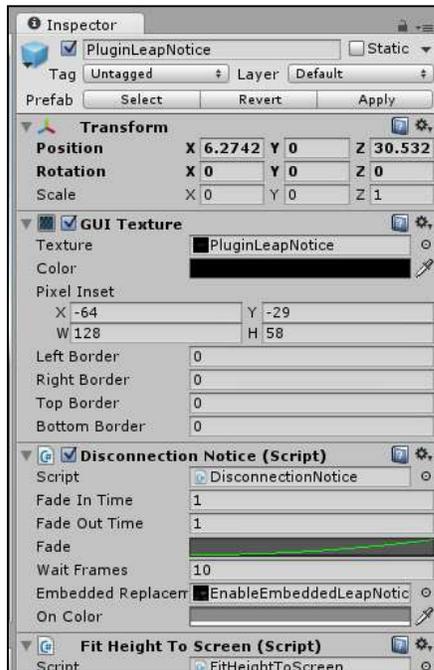


Figura 85. Script “PluginLeapNotice”

Gracias a este vínculo se puede utilizar el Hand Controller, que asigna la zona de interacción en la escena (donde aparecerán los modelos 3D de las manos) (figura 86) y el modelo físico de las manos con colisionadores.

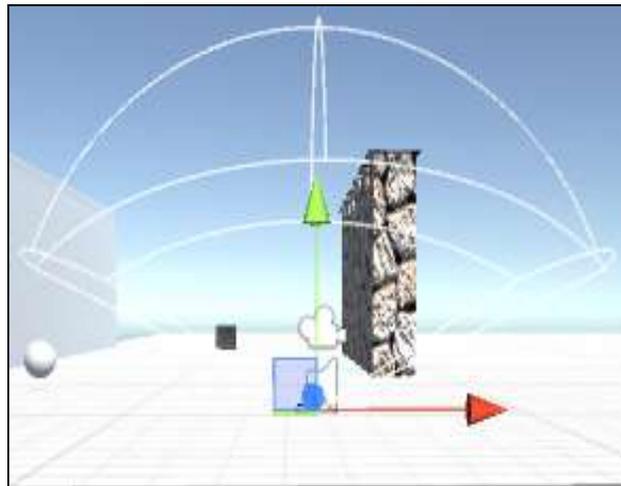


Figura 86. Script “Zona de interacción”

Los pasos anteriores bastarían para obtener una integración con el Leap Motion en una escena de Unity, sin embargo, en el presente trabajo se quería tener una interacción más fluida y quitar en la mayor medida posible el uso del ratón y teclado dentro de la aplicación por lo cual se agregó el siguiente prefabs:

- PinchingHand. Este facilita la toma de objetos dentro de un escenario y contiene todas las opciones de interacción de la mano, incluyendo el rango de apertura y cerrado ante los objetos, como se puede muestra en la figura 87.



Figura 87. Inspector de “PinchingHand”.

Una de las funciones que se configuran en el Leap Motion dentro del entorno, es el desplazamiento del personaje principal a través de las manos del usuario sin necesidad del uso de teclado o mouse.

Para poder navegar en el escenario se requirió de dos scripts, el primero “*mouse look*” que sirve para controlar la rotación de un objeto. Este script controla el movimiento de la cámara y la rotación del cuerpo del personaje principal controlado por el usuario (figura 88).

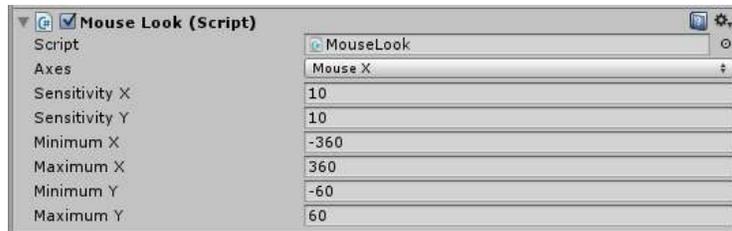


Figura 88. Script Mouse Look

Dentro de este Scripts se encuentran las opciones para configurar los ejes y ángulos de rotación, así como la sensibilidad de cada uno, gracias a los siguientes elementos:

- Axes: Sera en los ejes que tendrá la capacidad de controlar “Leap motion”, este puede ser x o y o ambos.
- Sensitivity x/y: Es la sensibilidad para detectar el movimiento de la mano y la velocidad del giro del objeto.
- Minimum x/y: mínimo ángulo de rotación.
- Maximum x/y: máximo ángulo de rotación.

A continuación se añade directamente en el cuerpo del personaje, para controlar el eje “x” (figura 89).

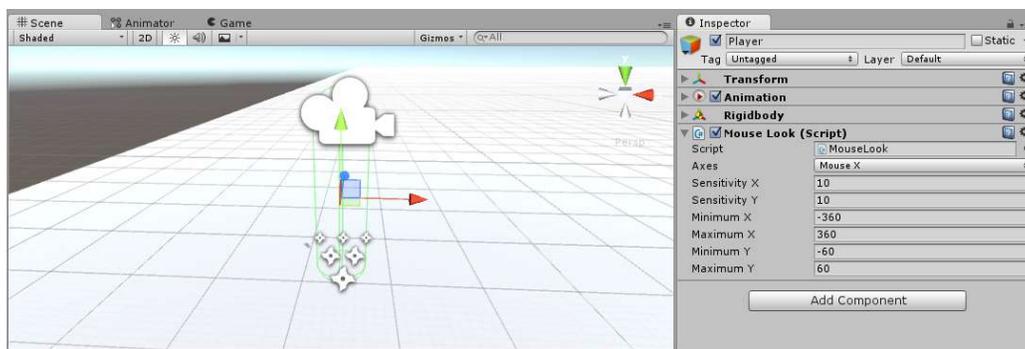


Figura 89. Script Mouse Look (cuerpo)

Posteriormente el script “mouse look” se agregó en la cámara, pero a diferencia del cuerpo se utilizó la opción de poder girar en ambos ejes de esta manera la rotación de la cámara y el cuerpo siempre se mantendrán sincronizados (figura 90).

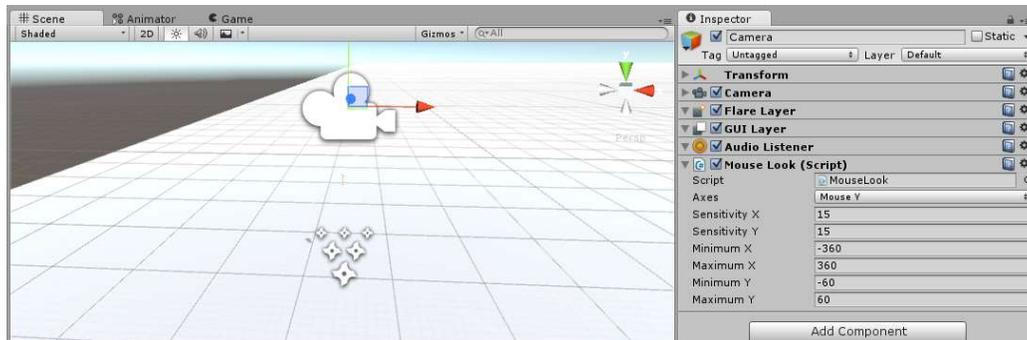


Figura 90. Script Mouse Look (cámara)

Obtenido el control de la cámara y la rotación del personaje, se adicionó el control del desplazamiento a través del script “LeapSuperHeroController” (figura 91) este se creó específicamente para el prefab del “constructor” que viene incluido en Unity al cual se le añadió una cámara en primera persona y deshabilito su MeshRender.”

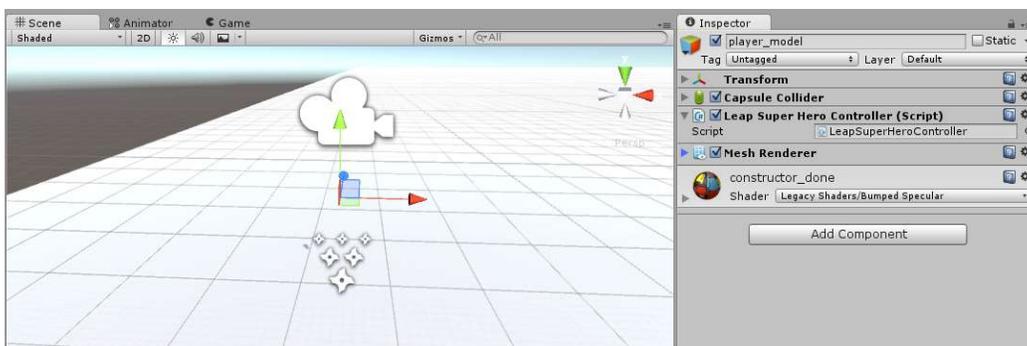


Figura 91. Script Mouse Look (cámara)

La unión de estos dos scripts quita la necesidad del uso del teclado y ratón en la aplicación, aún con esto scripts y prefabs, fue necesario hacer modificaciones directamente en el código para agilizar más la interacción con el usuario. Estas modificaciones serán explicadas en el siguiente apartado programación.

3.3.8 Programación

En este apartado se crearon los diferentes scripts y eventos necesarios para la interacción con el usuario y estabilidad del sistema.

3.3.5.1 Identificación de manos Leap motion

Los prefabs, son elementos prefabricados, los cuales son completamente funcionales, fáciles de usar y de modificar para ser incluidas en aplicaciones específicas. En este caso se modificaron los scripts de “Mouse Look” y “LeapSuperHeroControler”. En ellos se programaron (crearon) condiciones especiales para identificar cada mano del usuario por separado e integrar funciones en ellas.

El script “Mouse Look” se programó para ser usado solo por los movimientos de la mano izquierda, para lo cual se hizo una lista de las manos detectas por el sensor “Leap Motion” y en una variable se guarda la localización de la mano izquierda y derecha (figura 92).

```
40 //deteccion de manos
41
42     Frame frame = controller.Frame();
43     HandList hands = frame.Hands;
44     Hand firstHand = hands[0];
45     Hand hand = frame.Hands.Frontmost;
46 // separacion de mano izquierda y derecha
47
48     string handName = hand.IsLeft ? "Left hand" : "Right hand";
49
50     float xInput = Input.GetAxis("Mouse X");
51     float yInput = Input.GetAxis("Mouse Y");
52 --
```

Figura 92. Script Mouse Look detección de mano

Al final se crea la condición de solo tomar y detectar los movimientos de la palma izquierda debido a que esta controlara el movimiento de la cámara (figura 93).

```

66
67 // conteo de manos y deteccion de mano izquierda
68
69 if (frame.Hands.Count > 0 && hand.IsLeft) {
70     Vector3 palmPosition = frame.Hands[0].PalmPosition.ToUnityScaled();
71     Vector3 palmVelocity = frame.Hands[0].PalmVelocity.ToUnityScaled();
72     if (palmPosition.z > 0.0 && palmVelocity.magnitude < 8.0f && frame.Fingers.Count > 2) {
73         xInput = averageFingerVelocity.x * Mathf.Abs(averageFingerVelocity.x);
74         xInput *= 0.1f;
75         yInput = averageFingerVelocity.y * Mathf.Abs(averageFingerVelocity.y);
76         yInput *= 0.1f;
77     }

```

Figura 93. Script Mouse Look seguimiento de mano

En “LeapSuperHeroController” se cambió el ciclo de detección de “Mouse Look”, pero en este caso se priorizo la detección de la mano derecha (figura 94) para poder mover el personaje sin necesidad de teclado.

```

--
13 void LeapNavigation() {
14     Frame frame = m_leapController.Frame();
15     HandList hands = frame.Hands;
16     Hand firstHand = hands[0];
17     Hand hand = frame.Hands.Frontmost;
18     string handName = hand.IsLeft ? "Left hand" : "Right hand";
19
20     bool running = false;
21     if (frame.Hands.Count == 1 && hand.IsRight) {
22
23         Hand leftHand = frame.Hands[0];
24         Hand rightHand = frame.Hands[1];
25

```

Figura 94. Script LeapSuperHeroController detección de mano derecha

3.3.5.2 Reciclaje de componentes

Una de las características más importantes programadas en la presente aplicación, es el reciclaje de componentes para optimizar el uso de recursos de la computadora, esta función se programa en dos scripts.

El script destructor (“*destroy*”), fue programado para destruir los objetos que el usuario arroja cuando experimenta la caída libre y da pie a la creación de un nuevo objeto (figura 95).

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class Destroy2 : MonoBehaviour {
5
6     void OnCollisionEnter (Collision col)
7     {
8         if(col.gameObject.tag == "b")
9         {
10            Destroy(col.gameObject);
11        }
12    }
13 }
14

```

Figura 95. Script de destructor con condición

El script “instanciador” fue programado para crear nuevos objetos siempre y cuando haya sido marcado como destruido (figura 96).

```

Objetos 2
1 var oI:GameObject;
2 var nI: int;
3 var vG: int;
4 public var permiso: int = 0;
5 public var objeto : char;
6 private var rot: Quaternion = Quaternion.identity;
7 private var nInstancias:int = 0;
8 private var tiempoDeGeneracion: float;
9 function Update ()
10 {
11     if ((nInstancias <= nI) && (permiso == 1))
12     {
13
14         nInstancias +=1;
15         var objeto:GameObject = Instantiate(oI,gameObject.transform.position,rot);
16     }
17 }
18 function OnTriggerExit(otherObj: Collider){
19
20     if ( otherObj.tag == objeto)
21     {
22         permiso = 1;
23         nInstancias =0;
24         yield WaitForSeconds(0.1);
25         if ((nInstancias <= nI) && (permiso == 1))
26         {
27             var objeto:GameObject = Instantiate(oI,gameObject.transform.position,rot);
28         }
29         yield WaitForSeconds(0.1);
30         permiso = 0;
31         nInstancias =0;
32     }
33 }
34

```

Figura 96. Script de instanciador con condición

Una vez terminados los dos scripts, fueron colocados en objetos vacíos con “colisionadores” para detectar el paso de los objetos 3D a diferentes alturas de la torres, para que esta función se ejecute de manera automática durante la experimentación, (figura 97).

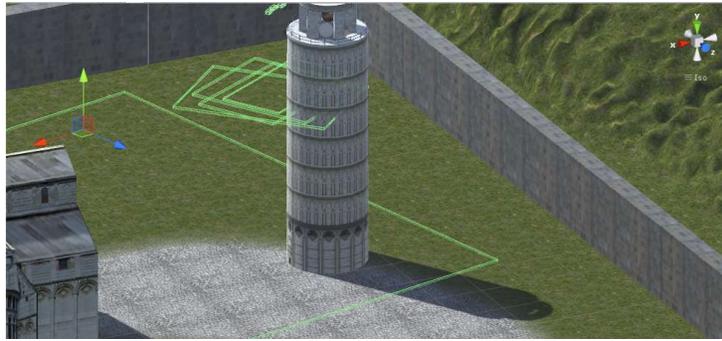


Figura 97. Colisionadores de reciclaje

3.3.5.3 Programación de gravedad

Una de las funciones prioritarias para que la aplicación se ejecute de manera adecuada y cumpla con los requisitos planteados es el control de la gravedad en cada escenario, el siguiente script (figura 98) permite recrear de manera realística la gravedad de los diferentes lugares donde se encontrara el usuario (cabe señalar que Unity cuenta con su propia opción para controlar la gravedad, pero no es tan precisa).

```
1 #pragma strict
2
3 var gravity = Vector3 (0, -9.81, 0);
4
5 function Awake () {
6     Physics.gravity = gravity;
7 }
```

Figura 98. Script de gravedad

3.3.5.4 Programación de cámaras

Otro requerimiento que cubrir es poder observar la caída de los objetos desde diferentes ángulos para que se compare el fenómeno con distintas variables, esto se logró programando un cambio de cámaras dentro del escenario (figura 99).

```

1 public var principal : Camera ;
2 public var secundaria : Camera ;
3
4
5 function OnTriggerEnter(otherObj: Collider){
6
7     if ( otherObj.tag == "B" || otherObj.tag == "V" || otherObj.tag == "U" ||otherObj.tag == "A")
8     {
9         principal.GetComponent.<Camera>().enabled = false;
10        secundaria.GetComponent.<Camera>().enabled = true;
11    }
12
13
14 }
15

```

Figura 99. Script cambio de camara

Las sentencias que se muestran en la figura 96 son las responsables del cambio de cámara, y para limitar su función se agregó una sentencia if, que se encarga de solo activar el cambio de cámara cuando un objeto pasa por el colisionador (figura 100).

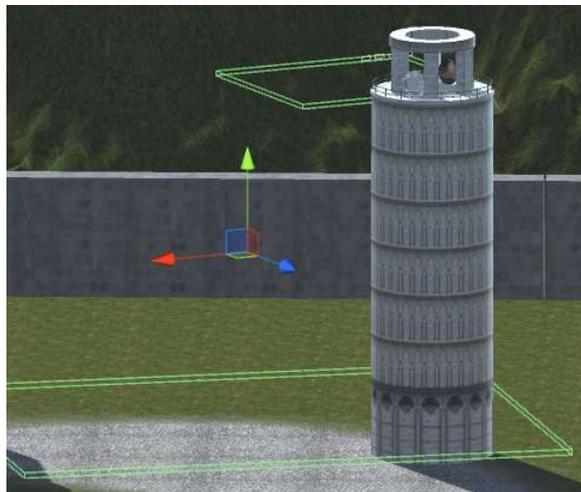


Figura 100. Colisionadores de cambio de cámara

A continuación, se crearon texturas con renderizado, este tipo de material reproduce las diferentes vistas de las cámaras para que sea posible ver varias al mismo tiempo. Esto permitió que la caída de los objetos se puedan observar desde varios ángulos en forma simultanea (figura 101).

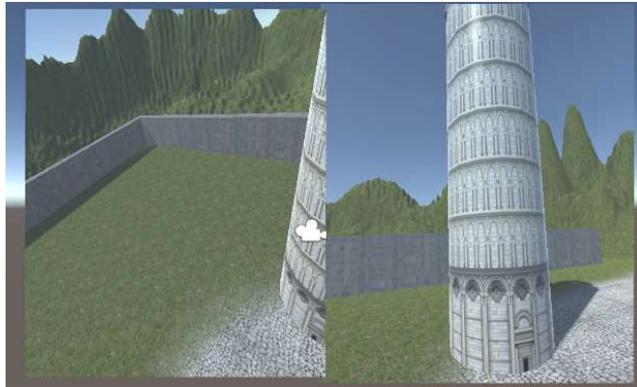


Figura 101. Texturas con render

Estas texturas se enlazaron directamente con las cámaras, para lograr el efecto (figura 102).

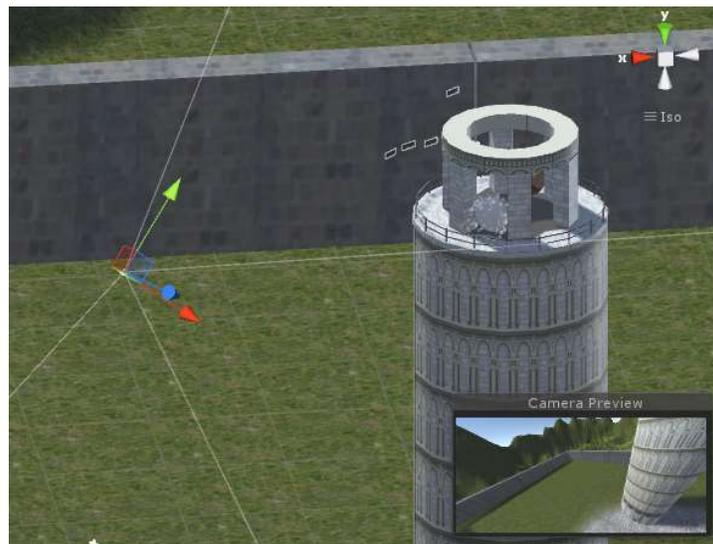


Figura 102. Cámaras de render.

3.3.6.5 Programación del tutorial de ayuda para el uso del Leap Motion

Se desarrolló un pequeño tutorial del uso del “*Leap motion*” dentro de los entornos de RV lo anterior para que el usuario se familiarice con su uso. Para esto se utilizaron habilitadores y destructores que liberan recursos conforme se cumplen las diferentes instrucciones del mismo (figura 103).

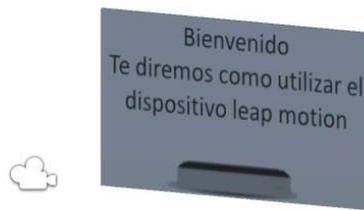


Figura 103. Tutorial leap.

En este script (figura 104) se enlistan todos los elementos necesarios (las diferentes camaras, las diferentes instrucciones, y objetos) para el desarrollo del tutorial, cada uno fue integrado en variables publicas para después ser utilizados según las condiciones que se van desencadenando. Cada condición es un paso que el usuario debe cumplir para progresar en el tutorial.

Una vez que el usuario termine podra iniciar de manera regular el recorrido dentro de la aplicacion.

```
Inspector Navigation
public var camleap : Camera ;
public var camplayer : Camera ;
public var leap : GameObject;
public var hand : GameObject;
public var mano : GameObject;
public var boton : GameObject;
public var objeto : GameObject;
public var deposito : GameObject;
public var barrera : GameObject;
public var canA : GameObject;
public var canB : GameObject;
public var canC : GameObject;
public var canD : GameObject;
public var instanciador : GameObject;

public var uso : int = 0;

function OnTriggerEnter(otherObj: Collider){
    if(otherObj.tag == "Untagged" && uso == 0)
    {
        Destroy (canA);
        canB.SetActive(true);
        yield WaitForSeconds(5);
        Destroy (canB);
        canC.SetActive(true);
        boton.SetActive(true);
        Destroy (mano);
    }
    if(otherObj.tag == "Untagged" && uso == 1)
    {
        Destroy (canA);
        Destroy (boton);
        canB.SetActive(true);
        objeto.SetActive(true);
        deposito.SetActive(true);
        instanciador.SetActive(true);
    }
    if(otherObj.tag == "A" && uso == 2)
    {
        Destroy (deposito);
        Destroy (objeto);
        Destroy (instanciador);
        Destroy (barrera);
        canA.SetActive(true);
    }
}
```

Figura 104. Script tutorial.

3.3.5.6 Programación de máquinas de estado y animación

Para crear una interacción entre el usuario y los distintos personajes fue necesario crear diferentes máquinas de estado, utilizando la herramienta “*animator*” de Unity, donde se agregan las animaciones junto con sus transiciones sirven como referencia para habilitar el siguiente estado (animación) como se puede ver en la figura 105.

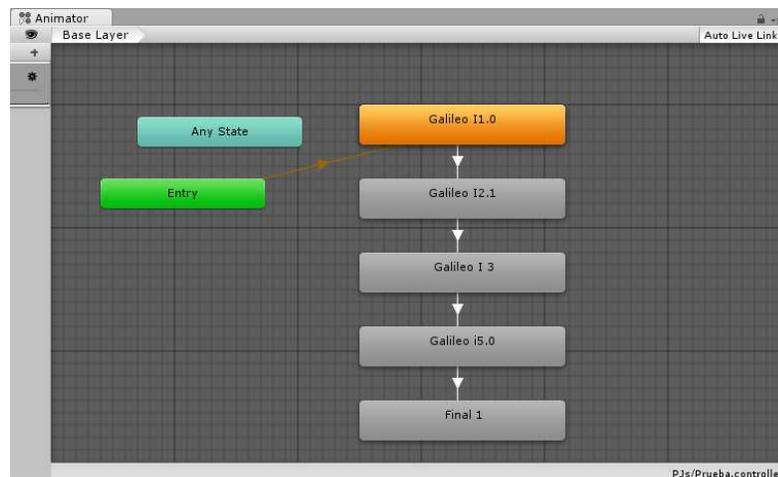


Figura 105. Máquina de estado

Para habilitar la transición se utilizó el siguiente script que contiene las variables para activar la secuencia de animación del personaje, este script sirve como interfaz entre el usuario y el personaje (figura 106).

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class amigoanim : MonoBehaviour {
5
6     public Animator _animamigo;
7     public bool _entreamigo;
8
9
10    void OnTriggerEnter (Collider other)
11    {
12
13        if (other.CompareTag ("Player"))
14        {
15            Debug.Log ("continuar");
16            _animamigo.SetBool ("continuar", true);
17        }
18    }
19
20    void OnTriggerExit (Collider other)
21    {
22        if (other.CompareTag ("Player"))
23        {
24
25            _animamigo.SetBool ("continuar", false);
26        }
27    }
28 }

```

Figura 106. Script de transición.

Estos scripts se enlazan con las transiciones de la máquina de estado y se programan diferentes If (condicionales), los cuales dependiendo de la secuencia que sea seleccionada por el usuario desencadenara las animaciones adecuadas para la interacción. Pero las secuencias de animación no contienen la voz del personaje y debe integrarse de manera manual.

Para sincronizar el audio con la animación debe crearse un acontecimiento en la pestaña “*animation*” de Unity donde se puede observar la secuencia junto con todos sus frames, para tomar como guía de donde debe iniciar la reproducción de voz y crear el evento “*animation event*” este será el encargado de reproducir el audio junto con la animación.



Figura 107. Eventos dentro la animación.

Para seleccionar la secuencia de audio es necesario integrar un script de audio como el que se muestra en la figura 108. Este se encarga de asignar el audio correcto para la animación y reproducirlo al activarse el “*animation event*”.

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class Audioal : MonoBehaviour
5 {
6     public AudioClip intro1;
7
8     // Use this for initialization
9     void Start()
10    {
11
12    }
13    void Playintro1()
14    {
15        GetComponent().clip = intro1;
16        GetComponent().Play();
17
18    }
19
20
21 }

```

Figura 108. Script de audio.

Este script se integra directamente en el inspector del personaje en Unity y se le asigna la secuencia de audio a reproducir.



Figura 109. Integración de audio

3.3.9 Pruebas de integración.

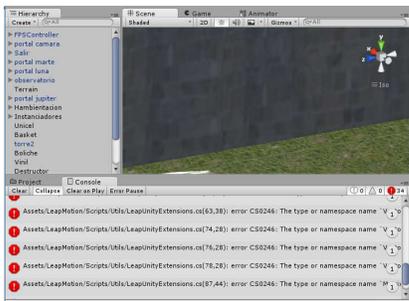
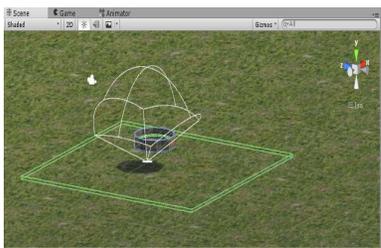
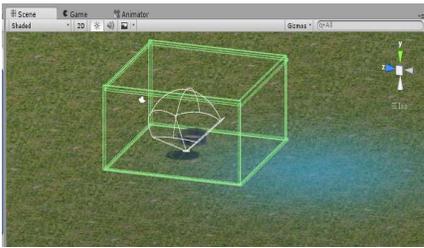
Para verificar el correcto funcionamiento de la aplicación y del dispositivo Leap Motion, constantemente se estuvo utilizando la opción ejecutar “Play” en la interface de *Unity* (figura 110). De esta manera se observó el comportamiento de los objetos y cuando fue necesario se hicieron correcciones pertinentes tanto de actualización de scripts, animaciones, objetos, destructores, etc.

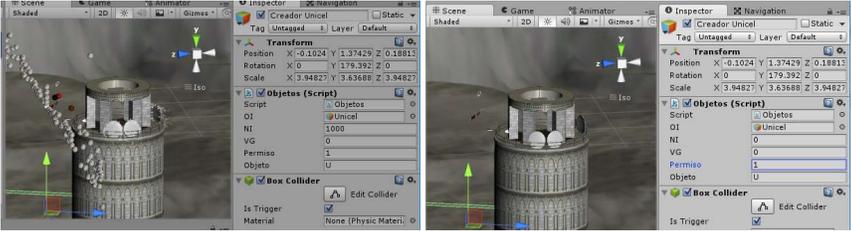
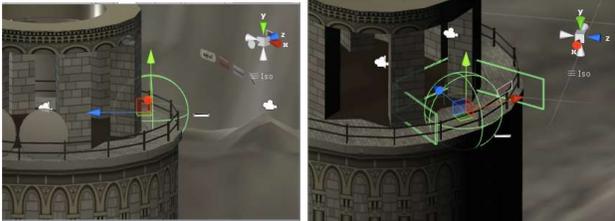
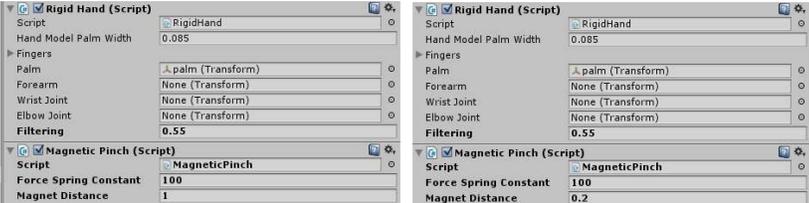
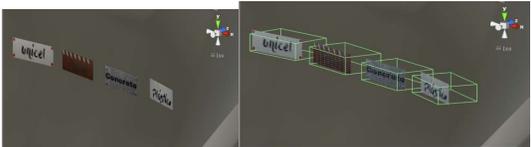


Figura 110. Ejecución de la aplicación

En la siguiente tabla se describen los errores detectados, así como las correcciones que se hicieron en ésta fase.

Tabla 20. Tabla de errores

Errores	Correcciones
<p>La aplicación no se puede conectar con Leap Motion</p>	<p>Este error se produjo por la actualización de la aplicación y de los “pluggins” de Leap Motion, se arregló actualizando la carpeta de “pluggins” dentro la aplicación de RV.</p>  
<p>Los objetos salen del alcance del usuario y el tutorial del dispositivo Leap Motion nunca termina.</p>	<p>El problema era que los objetos salían del rango de los destructores y nunca se detectaba su salida de la zona de interacción, se solucionó agregando una “caja de destrucción”.</p>  

<p>Mala reproducción de la animación.</p>	<p>Al reproducir la animación el personaje Galileo Galilei regresaba a las coordenadas 0 el problema se arregló agregando la opción “Root Motion”, el cual permite ejecutar la animación en cualquier parte del plano.</p> 
<p>Creación sin medida de objetos.</p>	<p>Al tener interacción los instanciadores se creó un error por no tener un límite en la cantidad máxima de objetos que se tendrá en cada escenario, por lo cual se ajustó la variable de objetos.</p> 
<p>Control de cámaras en ejecución.</p>	<p>Cuando se ejecutaba el regreso al escenario después de la experimentación el control del usuario quedaba atrapado en el colisionador de cambio de cámara y no permitía continuar con la aplicación, para solucionar esto se agregaron colisionadores que limitan la interacción con la experimentación.</p> 
<p>Control de distancia de agarre.</p>	<p>Cuando se ejecutaba la secuencia de experimentación la distancia para tomar objetos no es consistente lo que dificultaba la interacción, fue necesario cambiar el valor de la variable para una correcta ejecución.</p> 
<p>Colisionadores de aparición muy pequeños</p>	<p>Los colisionadores donde los objetos son instanciados estaban muy pequeños y los objetos se caían con facilidad, por lo cual se crearon colisionadores más grandes y mejor colocados para evitar que los objetos se cayeran mientras el usuario trata de tomarlos.</p> 

aplicación de RV, en esta misma sección se configura el sistema operativo en donde será desplegado y las arquitecturas soportadas (figura 112).

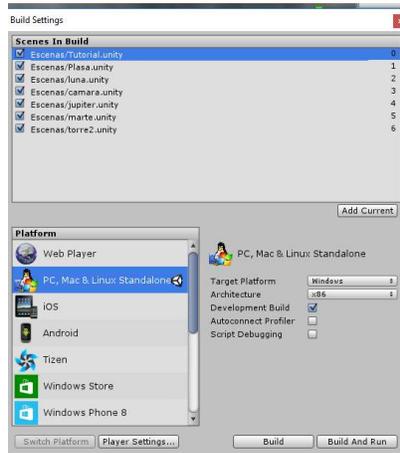


Figura 112. Opciones de sistema operativo

Para establecer un icono al archivo ejecutable de la aplicación que será visualizado en el archivo final, se abre el apartado “Player Settings” y se arrastra la imagen que se desea a la sección “Default Icon” (figura 113)

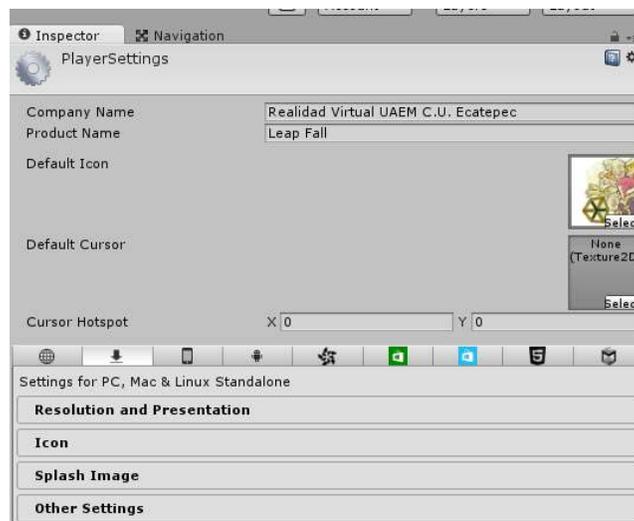


Figura 113. Selección de icono

Para determinar la resolución de despliegue, se selecciona “Resolution and Presentation” y se configuran las opciones “Default Screen Width” y el “Default Screen Height” en este

caso se utilizaron valores de 1280 por 800 para que el ancho de pantalla se el más adecuado para una buena representación (figura 114).

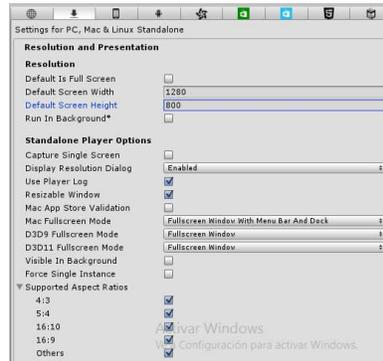


Figura 114. Selección de icono

Para finalizar se establece la ruta como destino del archivo ejecutable y se presiona “Build”, al finalizar el icono de la aplicación se muestra como la figura 115.



Figura 115. Empaquetado final

3.4 Fase de Evaluación y cierre

3.4.1 Valoración por el cliente / usuario.

Una vez que se consideró que la aplicación virtual para el estudio del movimiento unidimensional desarrollada estaba en versión Alfa, ya que cumplía con los requerimientos planteados al inicio del proyecto, se decidió utilizar el cuestionario propuesto por Cabero, Barroso y Llorente (2016) con la finalidad de obtener información desde la perspectiva del profesor que imparte el tema de caída libre en segundo grado de secundaria sobre el nivel de aceptación de la RA (Realidad Aumentada).

Este cuestionario está compuesto por 23 preguntas distribuidas en 7 escalas: 1) utilidad percibida, 2) facilidad de uso percibida, 3) percepción de disfrute, 4) actitud hacia el uso, 5) la intención de utilizar, 6) calidad del sistema y 7) competencia tecnológica. Contiene 4 reactivos socio demográficos.

Y tipo likert con siete opciones de respuesta 1) Extremadamente probable; 2) Muy probable; 3) Ligeramente probable; 4) Ninguno; 5) Poco probable; 6) Bastante improbable; 7) Extremadamente improbable.

Este cuestionario puede ser encontrado en Anexos.

Capítulo 4

Conclusiones

Se logró hacer una aplicación de RV inmersiva la cual puede ser utilizada dentro del conjunto de herramienta de apoyo didáctico empleadas para impartir el tema del fenómeno de caída libre dentro de una clase.

Al crear el presente proyecto con la metodología elegida se encontraron las siguientes oportunidades de mejora que permitirán agilizar el desarrollo de este tipo de aplicaciones:

- Crear un repositorio de contenido para almacenar los recursos artísticos que se vayan desarrollando, con el objetivo de que posteriormente puedan ser reutilizados en otras aplicaciones y a su vez minimizar tiempos de producción.
- Adecuar las fases de diseño y desarrollo al pipeline de videojuegos, para que se puedan desarrollar varias actividades de manera paralela, como: el diseño y desarrollo de los recursos artísticos (ej. Personajes) con la producción de los recursos tecnológicos (ej. Scripts para el funcionamiento de las máquinas de estado).

Por otra parte, después de que los profesores utilizaron la aplicación de RV, a través del cuestionario de aceptación tecnológica externaron estar capacitados para utilizar los entornos debido a que son fáciles de usar y manejar. Además, consideran que los recursos artísticos que los componen presentan una buena estructura, organización y funcionamiento.

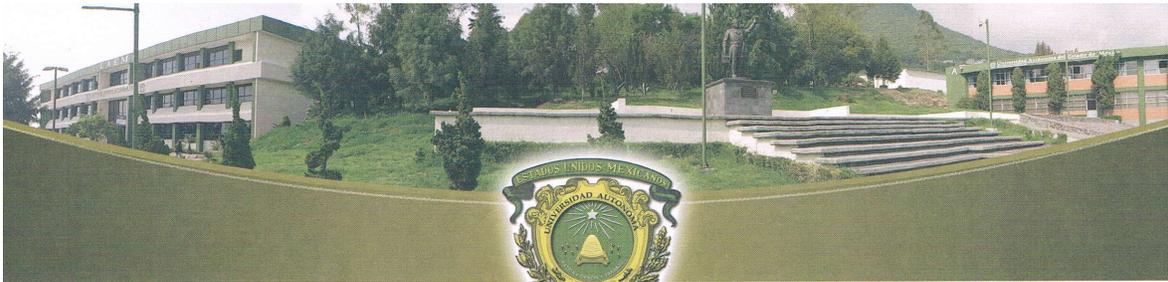
Aunado a lo anterior, evalúan en un rango de muy probable que la aplicación propicie la comprensión de los conceptos presentados, permita que los alumnos disfruten aprender y puedan reforzar los temas vistos en clase fuera del aula. Empero, la actitud hacia el uso la encuentran ligeramente probable, ya que consideran que quizás no es una buena idea incorporarla en la clase, aquí se tendría que reflexionar cuales son motivos del porque ésta respuesta, ya pueden afectar otros factores como son: que les genere a los profesores trabajo extra, no contar con el equipo adecuado, no tener permiso de alguna autoridad superior, etc.

Trabajos futuros

- Agregar la visualización de los datos de medición dentro de la interface del usuario, para poder el desarrollo matemático de los fenómenos.
- Incorporar más objetos y escenarios para tener más campos de experimentación del fenómeno.
- Incluir más fenómenos físicos para su experimentación (En especial más trabajos de Galileo Galilei).
- Incluir a mas figuras representativas de la física como actores dentro de la aplicación.

Anexos

Anexo 1



UAEM | Universidad Autónoma
del Estado de México
Centro Universitario UAEM Ecatepec
Otorga la presente
Constancia

A: LIC. JESÚS EDUARDO FLORES LARA

Por la Impartición del Taller Titulado:
“iSENSE with LEAP Motion”
en el Marco de la “Segunda Semana Multidisciplinaria”
llevada a cabo del 24 al 28 de Abril de 2017
en este Centro Universitario.

Ecatepec, Estado de México a 28 de Abril de 2017.

Patria, Ciencia y Trabajo

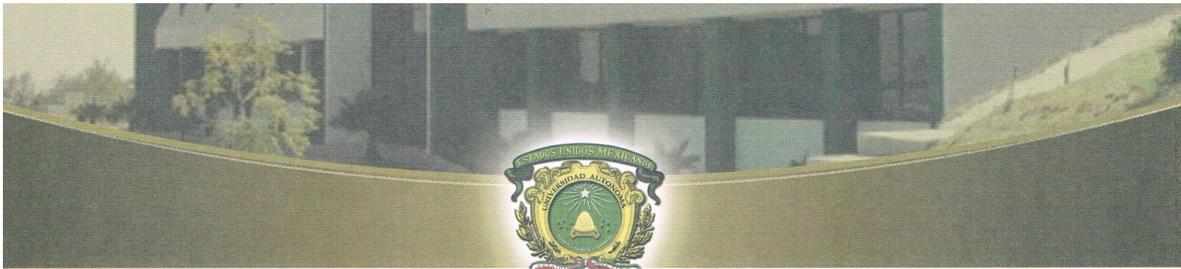
“2017, Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos”



[Handwritten Signature]
M. EN C. ED. MARCO ANTONIO VILLEDA ESQUIVEL
DIRECTOR DEL CENTRO UNIVERSITARIO UAEM ECATEPEC



Anexo 2



UAEM | Universidad Autónoma
del Estado de México

Centro Universitario Ecatepec

Otorga la presente

Constancia

A: *LIC. JESÚS EDUARDO FLORES LARA*

Por haber participado en la Conferencia:
Demostración de Realidad Virtual Inmersiva:
“Pon a Prueba tus Sentidos”
en el marco de la Semana Multidisciplinaria
llevada a cabo en este Centro Universitario.

Ecatepec, Estado de México a 19 de Abril de 2016.

Patria, Ciencia y Trabajo

“2016, Año del 60 Aniversario de la Universidad Autónoma del Estado de México”
“2016, Año de Leopoldo Flores Valdés”

M.en I.S.C. Cuauhtémoc Hidalgo Cortés
Encargado del Despacho de la Dirección CU UAEM Ecatepec

ECATEPEC



DIRECCIÓN



Anexo 3



UAEM | Universidad Autónoma
del Estado de México

Centro Universitario Ecatepec

Otorga la presente

CONSTANCIA

A: Flores Lara Jesús Eduardo

Por participar con la ponencia:

Aplicación de la tecnología de realidad virtual para el estudio del movimiento unidimensional: una actividad didáctica.

en el "Seminario de Investigación de la Maestría en Ciencias de Computación",
celebrado el día 02 de Junio del 2016 en las Instalaciones de este Centro
Universitario.

Patria, Ciencia y Trabajo

"2016, Año del 60 Aniversario de la Universidad Autónoma del Estado de México"
"2016, Año de Leopoldo Flores Valdés"

M. en I.S.C. Cuauhtémoc Hidalgo Cortés
Encargado del Despacho de la Dirección del CU UAEM Ecatepec



Anexo 4



UAEM | Universidad Autónoma
del Estado de México

La Sociedad Mexicana de Física y La Universidad Autónoma del Estado de México, a través del Centro Universitario UAEM Ecatepec, la Facultad de Ciencias en colaboración con el Gobierno del Estado de México, la Secretaría de Educación y el Consejo Mexiquense de Ciencia y Tecnología

Otorga la presente

CONSTANCIA

A: Flores Lara Jesús Eduardo

Por su participación como Jurado en la PRIMERA ETAPA del Concurso de Aparatos y Experimentos de Física, en el Estado de México.

Ecatepec de Morelos, Estado de México a 23 de mayo de 2015.

MTRO. CUAUHTÉMOC HIDALGO CORTÉS
ENCARGADO DEL DESPACHO DE LA DIRECCIÓN
DEL CENTRO UNIVERSITARIO UAEM ECATEPEC



GOBIERNO DEL
ESTADO DE MÉXICO



Anexo 5

Aumento de contenido de base de datos de laboratorio de realidad virtual.

Se agregaron diversos “skybox’s” al banco de datos del laboratorio de RV para su uso en futuros trabajos, así como algunos personajes, objetos y animaciones.

 animated_lava	29/05/2017 08:55 a...	Carpeta de archivos
 cstrike	29/05/2017 08:55 a...	Carpeta de archivos
 earthskybox	29/05/2017 08:55 a...	Carpeta de archivos
 flowsky08	29/05/2017 08:55 a...	Carpeta de archivos
 galaxy_2	29/05/2017 08:55 a...	Carpeta de archivos
 hd_metal	29/05/2017 08:55 a...	Carpeta de archivos
 jupiter	29/05/2017 08:55 a...	Carpeta de archivos
 land004_2	29/05/2017 08:55 a...	Carpeta de archivos
 lava	29/05/2017 08:55 a...	Carpeta de archivos
 Mars skybox 4	29/05/2017 08:55 a...	Carpeta de archivos
 mars1_2	29/05/2017 08:55 a...	Carpeta de archivos
 metal_textures_3	29/05/2017 08:55 a...	Carpeta de archivos
 mpa98	29/05/2017 08:55 a...	Carpeta de archivos
 mpa120	29/05/2017 08:55 a...	Carpeta de archivos
 neon_textures_v2	29/05/2017 08:55 a...	Carpeta de archivos
 otherworld	29/05/2017 08:55 a...	Carpeta de archivos
 Out of mars Skybox	29/05/2017 08:55 a...	Carpeta de archivos
 pebblesgrassblend	29/05/2017 08:55 a...	Carpeta de archivos
 skybox_pack	29/05/2017 08:55 a...	Carpeta de archivos
 stonewall	29/05/2017 08:58 a...	Carpeta de archivos
 winterplanet skybox	29/05/2017 08:58 a...	Carpeta de archivos

Anexo 6

Instrumento para medir el nivel de aceptación de la RV.

Genero	F	M	Edad							
Área curricular			Nivel de estudios							
Instrucciones	<p>El cuestionario es anónimo y voluntario; sus respuestas se mantendrán confidenciales, por lo que se solicita su opinión sincera al respecto. No hay respuestas buenas o malas; todas son valiosas. Responda después de leer cuidadosamente cada enunciado. Debe responder en la casilla de la letra correspondiente (A, B, C, D, E, F o G).</p> <p>A) extremadamente probable; B) muy probable; C) ligeramente probable; D) ninguno; E) poco probable; F) bastante improbable y G) extremadamente improbable</p>									
Número	Pregunta			A	B	C	D	E	F	G
1	El uso de este sistema de RA mejorará mi aprendizaje y rendimiento en esta asignatura (UP1).									
2	El uso del sistema de RA durante las clases me facilitaría la comprensión de ciertos conceptos (UP2).									
3	Creo que el sistema de RA es útil cuando se está aprendiendo (UP3).									
4	Con el uso de la RA aumentaré mi rendimiento (UP4).									
5	Creo que el sistema de RA es divertido (FUP1).									
6	Aprender a usar el sistema de RA no es un problema para mí (FUP2).									
7	Aprender a usar el sistema de RA es claro y comprensible (FUP3).									
8	Utilizar el sistema de RA es divertido (DP1).									
9	Disfruté con el uso del sistema de RA (DP2).									
10	Creo que el sistema de RA permite aprender jugando (DP3).									
11	El uso de un sistema de RA hace que el aprendizaje sea más interesante (AU1).									
12	Me he aburrido utilizando el sistema de RA (AU2).									
13	Creo que el uso de un sistema de RA en el aula es buena idea (AU3).									
14	Me gustaría utilizar en el futuro el sistema de RA si tuviera oportunidad (IU1).									
15	Me gustaría utilizar el sistema de RA para aprender anatomía como otros temas (IU2).									
16	.El funcionamiento del recurso de RA que te hemos presentado es:									
17	.En general, la estética del recurso producido en RA lo consideras:									

18	En general, el funcionamiento técnico del recurso producido en RA lo calificaría de:								
19	En general, cómo valorarías la presentación de la información en la pantalla:								
20	Los objetos producidos en RV son seguros.								
21	De manera general me considero capacitado para el manejo técnico de los medios audiovisuales e informática.								
22	De manera general me considero capacitado para el manejo técnico de Internet.								
23	De manera general me considero capacitado para el manejo técnico de los objetos producidos en el sistema de RV.								

Una vez analizado el instrumento, se modificó la redacción de algunas preguntas con el objetivo de que fuera más claras para los profesores, dejando la mayoría de los aspectos y la escala intactos.

Tipo de plantilla	Instrumento para medir el nivel de aceptación de la aplicación en los alumnos desde el punto de vista del profesor.									
Genero	F	M	Edad							
Área curricular			Nivel de estudios							
Instrucciones	<p>El cuestionario es anónimo y voluntario; sus respuestas se mantendrán confidenciales, por lo que se solicita su opinión sincera al respecto. No hay respuestas buenas o malas; todas son valiosas. Responda después de leer cuidadosamente cada enunciado. Debe responder en la casilla de la letra correspondiente (A, B, C, D, E, F o G).</p> <p>A) extremadamente probable; B) muy probable; C) ligeramente probable; D) ninguno; E) poco probable; F) bastante improbable y G) extremadamente improbable</p>									
Número	Pregunta			A	B	C	D	E	F	G
1	El uso de la aplicación de RV podría mejorar el aprendizaje de los estudiantes en el aula.									
2	La utilización de la aplicación de RV durante la clase facilitaría a los estudiantes la comprensión de ciertos conceptos.									
3	Creo que el uso de las aplicaciones de RV es útil cuando se está enseñando y los estudiantes están aprendiendo.									
4	Creo que la aplicación de RV es fácil de usar.									
5	Aprender a usar y manejar la aplicación de RV no es un problema para mí.									
6	La interacción con la aplicación de RV es clara y comprensible.									
7	Creo que la aplicación de RV permite que los estudiantes refuercen el tema visto en clase jugando.									
8	Disfruté la clase con el uso de la aplicación de RV.									

9	La clase es entretenida con una aplicación de RV de este tipo.								
10	El uso de la aplicación de RV hace que la clase sea más interesante.								
11	Enseñar el tema de caída libre a mis estudiantes a través de la aplicación de RV me pareció aburrido.								
12	Creo que el uso de un sistema de RV en el aula es una buena idea								
13	Me gustaría utilizar la aplicación de RV en el futuro si tuviera la oportunidad.								
14	El uso de aplicaciones de RV les permitiría a mis alumnos repasar y reforzar los temas vistos en clase por su cuenta.								
15	Me gustaría utilizar aplicaciones de RV para que mis estudiantes repasen y refuercen los temas de clase								
16	Los objetos producidos en la aplicación de RV son estéticamente adecuados para el tema en cuestión.								
17	Los objetos producidos en la aplicación de RV funcionan correctamente								
18	Los objetos producidos en la aplicación de RV están bien estructurados y organizados								
19	Los objetos producidos en la aplicación de RV son atractivos.								
20	Los objetos producidos en la aplicación de RV son seguros.								
21	De manera general me considero capacitado para el manejo técnico de los objetos producidos en la aplicación de RV								

I. Bibliografía

- Antonietti, A., Rasi, C., Imperio, E., & Sacco, M. (2000). The representation of virtual reality in education. *Education and Information Technologies*, 5(4), 317-327. DOI: 10.1023/A: 1012057608694
- Dede, C., Salzman, M., Loftin, R. B., & Ash, K. (1997). Using virtual reality technology to convey abstract scientific concepts. *Learning the Sciences of the 21st Century: Research, Design, and Implementing Advanced Technology Learning Environments*. Lawrence Erlbaum: Hillsdale, NJ.
- Dickey, M. D. (2005). Three-dimensional virtual worlds and distance learning: two case studies of Active Worlds as a medium for distance education. *British journal of educational technology*, 36(3), 439-451. DOI: 10.1111/j.1467-8535.2005.00477.x
- Guillén, D. Z. (2011). La realidad virtual como recurso y herramienta útil para la docencia y la investigación. *TE & ET: Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (6), 17-23.
- Gurău, C. (2007). Virtual Reality Applications in Tourism. En Pease, W., Rowe, M. & Cooper, M. (Ed.), *Information and Communication Technologies in Support of the Tourist Industry* (180-197). USA: IGI Global. DOI: 10.4018/978-1-59904-955-7.ch054.
- Hui-Zhen, R. & Zong-Fa, L. (2013). Application and Prospect of the Virtual Reality Technology in College Ideological Education. *In Intelligent Systems Design and Engineering Applications, 2013 Fourth International Conference on*. DOI: 10.1109/ISDEA.2013.434.
- Ibanez, M.B., Garcia, J.J., Galan, S., Maroto, D., Morillo, D. & Kloos, C.D. (2011). Design and Implementation of a 3D Multi-User Virtual World for Language Learning. *Educational Technology & Society*, 14(4), 2-10. DOI: 10.1080/09687760802526707.

- Kastoudi, D. (2011). Using a Quest in a 3D Virtual Environment for Student Interaction and Vocabulary Acquisition in Foreign Language Learning. *European Association for Computer-Assisted Language Learning (EUROCALL)*, 20, 87-89.
- Kaufmann, H., & Schmalstieg, D. (2006). Designing immersive virtual reality for geometry education. In *Virtual Reality Conference, 2006* (51-58). IEEE. DOI: 10.1109/VR.2006.48.
- Knutzen, K. B., & Kennedy, D. M. (2012). Designing the self: the transformation of the relational self-concept through social encounters in a virtual immersive environment. *Interactive Learning Environments*, 20(3), 271-292.
- Pérez, M., Zabre E. & Islas E. (2004). Realidad Virtual: Un panorama general. *Boletín IIE, Instituto de Investigación Electrónicas*, 28 (2), 39-44.
- Rahimi, A., Golshan, N. & Mohebi, H. (2013). Virtual Reality as a Learning Environment in Iranian EFL Context: Personal, Technical, and Pedagogical, *Procedia - Social and Behavioral Sciences*, 136, 234 – 239. DOI: 10.1016/j.sbspro.2014.05.320.
- Shi, Y. L. (2014). Application of Virtual Reality Technology in Medical Education. En Li, S., Jin, Q., Jiang, X. & Park, J. (Ed), *Frontier and Future Development of Information Technology in Medicine and Education* (467-476). Netherlands: Springer. DOI: 10.1007/978-94-007-7618-0_46.
- Pantelidis, V. Reasons to Use Virtual Reality in Education. Disponible en: <http://vr.coe.edu/reas.html>. Consultado el 08 de diciembre de 2014.
- Valdivia, A. O. (2003). El construccionismo y sus repercusiones en el aprendizaje asistido por computadora. *Contactos*, 48, 61-64.
- Zhan, Z. (2011). The Application of VR on Distance Education. En Chen, R. (Ed.), *Intelligent Computing and Information* (78-83). Berlin Heidelberg: Springer. DOI: 10.1007/978-3-642-18134-4_13.

- R. Hui-Zhen and L. Zhong-Fa, (2013). Application and Prospect of the Virtual Reality Technology in College Ideological Education. Fourth International Conference on Intelligent Systems Design and Engineering Applications.
- F. Esteve, J. Segura and M. Gisbert, “Diseño de un entorno para el desarrollo de la competencia digital docente en estudiantes universitarios: usabilidad, adecuación y percepción de utilidad,” RELATEC, vol 13(2), pp. 35-47, 2014.
- E. Shudayfat, F. Moldoveanu and A. Moldoveanu, “A 3D virtual learning environment for teaching chemistry in high school,” Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, vol. 23 (1) pp. 423-428, 2012.
- A. Morales, L. E. Alviter, C. Hidalgo, J. J. Amador, J.A. Zúñiga, et al., “Metodología de Desarrollo Evolutivo de Escenarios Tridimensionales para la Creación de una Visita Virtual para el Centro Universitario UAEM Ecatepec,” Revista Colombiana de Computación, vol. 16 (1), pp. 7-17, 2015