

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE GRADO

SOLUCIÓN DE BAJO COSTE DE CAPTURA DE MOVIMIENTO BASADA EN KINECT

Titulación: Grado en Ingeniería en Tecnologías de la Telecomunicación

Autor: Néstor Manuel Alemán Soler

Tutores: Dionisio Rodríguez Esparragón

Luis Domínguez Quintana

Fecha: Julio 2014

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE GRADO

SOLUCIÓN DE BAJO COSTE DE CAPTURA DE MOVIMIENTO BASADA EN KINECT

HOJA DE FIRMAS

Alumno/a

Fdo.: Néstor Manuel Alemán Soler

Tutor/a

Tutor/a

Fdo.: Dionisio Rodríguez Esparragón

Fdo.: Luis Domínguez Quintana

Fecha: Julio 2014

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE GRADO

SOLUCIÓN DE BAJO COSTE DE CAPTURA DE MOVIMIENTO BASADA EN KINECT

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Fdo.:

Vocal

Fdo.:

Secretario/a

Fdo.:

Fecha: Julio 2014

Índice de contenidos

CAPÍTULO 1. INTRODUCCIÓN

1.1 ANTECEDENTES	15
1.2 OBJETIVOS.....	17
1.3 ESTRUCTURA DE LA MEMORIA	17

CAPÍTULO 2. CAPTURA DE MOVIMIENTO

2.1 ESTADO DEL ARTE	21
2.1.1 <i>Historia de la captura de movimiento</i>	21
2.2 TECNOLOGÍAS DE CAPTURA DE MOVIMIENTO	24
2.3 CAPTURA DE MOVIMIENTO CON KINECT	30

CAPÍTULO 3. SOFTWARE DE DISEÑO 3D

3.1 SOFTWARE PARA EL DISEÑO 3D	35
3.1.1 <i>Software comercial</i>	35
3.1.2 <i>Software libre</i>	38
3.2 BLENDER.....	42
3.2.1 <i>Características principales de Blender</i>	42
3.2.2 <i>Blender y la captura de movimiento</i>	43

CAPÍTULO 4. DESARROLLO DEL TUTORIAL

4.1 OBJETIVOS	47
4.2 DISEÑO DEL TUTORIAL	47
4.3 MATERIAL UTILIZADO	48
4.3.1 <i>Software</i>	48
4.3.2 <i>Hardware</i>	49
4.4 PROCESO DE REALIZACIÓN DEL TUTORIAL	49
4.4.1 <i>Fase 1.Estructura</i>	49
4.4.2 <i>Fase 2. Formato</i>	50
4.4.3 <i>Fase 3. Producción del contenido</i>	52
4.4.4 <i>Fase 4. Material de apoyo</i>	56

CAPÍTULO 5. PRUEBAS Y CONCLUSIONES

5.1 PRUEBAS REALIZADAS.....	65
5.1.1 <i>Configuración de la iluminación del entorno</i>	65
5.1.2 <i>Campo y entorno visual</i>	68
5.2 CONCLUSIONES	72

BIBLIOGRAFÍA73

PRESUPUESTO77

ANEXO I87

Índice de figuras

FIGURA 2.1. EL CABALLO EN MOVIMIENTO DE EADWEARD MUYBRIDGE.	21
FIGURA 2.2. EJEMPLO DE USO DE LA TÉCNICA DE LA ROTOSCOPIA UTILIZADA EN LA PELÍCULA DE BLANCANIEVES Y LOS 7 ENANITOS DE DISNEY.	22
FIGURA 2.3. MIKE LA CABEZA PARALANTE, A LA DERECHA. A LA IZQUIERDA, EL ARTISTA CON EL DISPOSITIVO QUE LO CONTROLABA	23
FIGURA 2.4. MOXY.	23
FIGURA 2.5. EJEMPLO DE UN SISTEMA DE CAPTURA DE MOVIMIENTO ELECTROMECAÁNICA.....	24
FIGURA 2.6. EJEMPLO DE UN SISTEMA DE CAPTURA DE MOVIMIENTO ELECTROMAGNÉTICA.	25
FIGURA 2.7. EJEMPLO DE UN SISTEMA DE CAPTURA DE MOVIMIENTO ÓPTICA MEDIANTE INDICADORES PASIVOS.....	27
FIGURA 2.8. EJEMPLO DE UN SISTEMA DE CAPTURA DE MOVIMIENTO ÓPTICA MEDIANTE INDICADORES ACTIVOS.	28
FIGURA 2.9. IMAGEN DEL KINECT DE MICROSOFT.....	31
FIGURA 3.1. MUESTRA DEL SOFTWARE 3DS MAX.	35
FIGURA 3.2. MUESTRA DEL SOFTWARE AUTODESK MAYA.	36
FIGURA 3.3. MUESTRA DEL SOFTWARE CINEMA 4D	37
FIGURA 3.4. MUESTRA DEL SOFTWARE GOOGLE SKETCHUP.	38
FIGURA 3.5. MUESTRA DEL SOFTWARE WINGS 3D.	39
FIGURA 3.6. MUESTRA DEL SOFTWARE K-3D.	40
FIGURA 3.7. MUESTRA DEL SOFTWARE BLENDER.	41
FIGURA 3.8. EJEMPLO DE MODELO RENDERIZADO EN BLENDER.	43
FIGURA 4.1. PARTES EN LAS QUE SE DIVIDE EL TUTORIAL.	49
FIGURA 4.2. MUESTRA DEL FORMATO EMPLEADO EN EL TUTORIAL.....	51
FIGURA 4.3. MUESTRA DE LA INTERFAZ DEL SOFTWARE FRAPS.	53
FIGURA 4.4. MUESTRA DE LA HERRAMIENTA RECORTES DE WINDOWS 7.....	53
FIGURA 4.5. EJEMPLO DE CAPTURA SIN RETOQUE FOTOGRÁFICO.....	54
FIGURA 4.6. EJEMPLO DE LA CAPTURA YA RETOCADA.	54
FIGURA 4.7. EJEMPLO DE UNA CAPTURA SIN SEÑALIZACIÓN.....	55
FIGURA 4.8. EJEMPLO DE UNA CAPTURA CON SEÑALIZACIÓN.	55
FIGURA 4.9. MUESTRA DE LAS CARPETAS QUE SE ADJUNTAN CON EL TUTORIAL.	56
FIGURA 4.10. COMPROBACIÓN DE UN ESQUELETO.....	57
FIGURA 4.11. MUESTRA DEL PROCESO SEGUIDO PARA LA REALIZACIÓN DEL VIDEO DEMOSTRATIVO.	57
FIGURA 4.12. ORIENTACIÓN DE LA KINECT.	58
FIGURA 4.13. PROCESO DE CALIBRACIÓN DE LA KINECT.	59
FIGURA 4.14. SET DE ILUMINACIÓN.	60
FIGURA 4.15. MUESTRA DE MOMENTO DE EDICIÓN, DONDE SE OBSERVA LA DIVISIÓN DE LA PANTALLA CON LAS DOS SEÑALES DE VIDEO UNIDAS.	61
FIGURA 5.1. CONFIGURACIÓN DEL PLATÓ PARA LA REALIZACIÓN DE LA CAPTURA DE MOVIMIENTO.	65
FIGURA 5.2. PRUEBA DEL KINECT BAJO LA LUZ DE 3 FOCOS.	66
FIGURA 5.3. PRUEBA DEL KINECT BAJO LA LUZ DE 1 FOCO DE LUZ.....	67
FIGURA 5.4. PRUEBA DEL KINECT CON LUZ NATURAL.	67
FIGURA 5.5. PRUEBA EN LA QUE EL INDIVIDUO SE ACERCA DEMASIADO AL DISPOSITIVO KINECT.....	69
FIGURA 5.6. PRUEBA EN LA QUE EL INDIVIDUO SE ACERCA DEMASIADO AL DISPOSITIVO KINECT, PERO ESTA VEZ CON EL FONDO DE COLOR BLANCO. ANALIZANDO SI ES IMPORTANTE QUE HAYA UN CONTRASTE ENTRE EL FONDO Y LA ROPA DEL INDIVIDUO.	70
FIGURA 5.7. PRUEBA DEL INDIVIDUO EN CONTACTO CON LA PARED DE FONDO BLANCO.	71
FIGURA 5.8. PRUEBA DEL INDIVIDUO EN CONTACTO A LA PARED DE FONDO VERDE.	71

Índice de tablas

TABLA P1. RECURSOS HARDWARE.	81
TABLA P2. RECURSOS SOFTWARE.	82
TABLA P3. FACTOR DE CORRECCIÓN SEGÚN EL COIT.	83
TABLA P4. PRESUPUESTO TOTAL HASTA EL MOMENTO.....	84
TABLA P5. MATERIAL FUNGIBLE.	85
TABLA P6. COSTE TOTAL DEL TRABAJO.	86

Capítulo 1.

Introducción

En este capítulo se detalla el propósito de este Trabajo Fin de Grado, y se describen los objetivos que se desean conseguir con el mismo. Para finalizar, se realiza un resumen de la estructura de la memoria de este Trabajo de Fin de Grado, detallando brevemente capítulo a capítulo.

1.1 Antecedentes

En los últimos 20 años, sobre todo en el mundo del cine y de los videojuegos, se han desarrollado nuevas técnicas de animación. El objeto de estos desarrollos es obtener una animación más realista.

Sin existir ningún método de animación automático perfecto, una de las técnicas que mejores resultados da es la captura de movimiento, o MOCAP. El MOCAP consiste en obtener las posiciones corporales de actores para animar los modelos digitales de personajes en 2D y 3D. Tal y como se ha dicho, esta técnica se utiliza para conseguir que los movimientos de los personajes sean lo más reales posibles, logrando, además, un ahorro en el volumen de trabajo de los animadores.^[1]

Sin embargo, estos sistemas de captura de movimiento, basados en hardware y software dedicados, son muy costosos y sólo están disponibles en estudios que trabajen específicamente en proyectos de animación.

Como alternativa a los sistemas hardware, existen referencias en la bibliografía de proyectos de desarrollo basados en la integración de soluciones de bajo coste empleando hardware como Kinect para la detección y captura de movimiento. La mayoría de estos proyectos están orientados al uso de los datos de captura con software de modelado y animación de pago, tipo 3DS MAX, MAYA o SOFTIMAGE, o aplicaciones propietarias del desarrollador. Sin embargo, existen soluciones software de modelado y animación muy potentes basadas en desarrollos libres. Es el caso de Blender, una de las herramientas de modelado basada en código abierto más populares del mundo.

Con este trabajo lo que se pretende es estudiar e implementar una solución de bajo coste basada en Kinect para la captura de movimiento y su integración con Blender, de manera que se cuente con una solución eficiente para la realización de captura de movimiento en animación. Dicho desarrollo será apoyado con la elaboración de un tutorial sobre el procedimiento necesario para realizar MOCAP con Kinect y Blender, facilitando así su uso con fines formativos.

Kinect, originalmente conocido como Project Natal, es un dispositivo que detecta el movimiento de las personas en tiempo real siendo capaz de reconocer gestos y comandos por voz. Kinect dispone de una cámara de color y de otras dos que detectan la profundidad. De esta forma es posible calcular dónde hay un cuerpo así como las diversas articulaciones de éste para poder recrear el esqueleto.^[2]

Algunos ejemplos del uso de Kinect como dispositivo de captura de movimiento son los proyectos: Brekel Kinect, creado por Jasper Brekelmans, que guarda la información sobre una animación en tiempo real en un formato de texto para poder usarla más adelante para animar un modelo 3D^[3] o el proyecto Kinect Hacking, que tiene como objetivo capturar el movimiento con Kinect de una persona y traspasarlo directamente al personaje que controlamos en un videojuego. De la misma manera, el dispositivo Kinect también se emplea en otros campos como la medicina para el tratamiento de pacientes que padecen de esclerosis múltiple^[4], diagnóstico de desórdenes infantiles^[5], rehabilitación^[6] o teleasistencia^[7].

Blender, originalmente desarrollado en 1995 por Ton Roosendaal, cofundador del estudio de animación holandés NeoGeo^[8], es un software que ofrece un amplio espectro de herramientas para el modelado, adición de texturas, iluminación, animación y funcionalidad post-procesamiento de vídeo en un solo paquete. Gracias a su arquitectura abierta, Blender proporciona interoperabilidad entre plataformas, extensibilidad, y un flujo de trabajo totalmente integrado que lo han convertido en un referente para la comunidad.^[9]

Algunas aplicaciones del software en la industria audiovisual son la integración de efectos especiales en cine y televisión^[10], cine de animación 3d^[11], videojuegos^[12], etc.

1.2 Objetivos

Los dos objetivos principales que se pretenden conseguir en este Trabajo Fin de Grado son los siguientes:

- Implementar un sistema de captura de movimiento de bajo coste, usando, como dispositivo de captura, el Kinect de Microsoft.
- Implementar la solución MOCAP con el software de animación Blender, así como elaborar un tutorial del uso de Blender para este propósito.

1.3 Estructura de la memoria

A continuación se describe brevemente el contenido de cada uno de los capítulos que conforman este Trabajo Fin de Grado.

- **Capítulo 2. Captura de Movimiento.**

En este capítulo se hace una introducción a la captura de movimiento, así como una breve descripción de distintos tipos de sistemas de captura de movimiento. También se describe el dispositivo que se ha utilizado para este trabajo, el Kinect.

- **Capítulo 3. Software de diseño 3D**

En este capítulo se tratan los softwares de diseño 3D y las diferentes clases que hay en el mercado. A su vez, se hace una descripción del software escogido para este trabajo, el Blender.

- **Capítulo 4. Desarrollo del tutorial**

En este capítulo se detallan los objetivos del tutorial, objeto del proyecto y, además, se expone el proceso para la realización del mismo.

- **Capítulo 5. Pruebas y conclusiones**

En este capítulo se especifican las pruebas realizadas del sistema implementado, así como los resultados obtenidos. Además se revisarán los objetivos y se extraerán las diferentes conclusiones y proponiendo posibles mejoras para el futuro.

Capítulo 2.

Captura de movimiento

Una vez estudiados los objetivos a desarrollar, se hará una introducción a la captura de movimiento, además de una breve descripción de los distintos tipos de sistemas de captura de movimiento. Por último se describirá el dispositivo que se ha utilizado para este trabajo, el Kinect.

2.1 Estado del arte

La captura de movimiento (Motion Capture, en inglés) es el proceso de digitalizar movimientos reales de objetos o personas y añadir, dichos movimientos, a modelos 3D. Actualmente, en el cine se usa mucho esta técnica, ya que facilita el trabajo de los animadores. Otro campo en el que esta técnica es puntera, sin duda, es en el de los videojuegos. Para ofrecer una experiencia más realista a los usuarios, se utilizan técnicas de captura de movimiento para naturalizar los movimientos de los personajes de los juegos.

La técnica de animación de personajes empleando la captura de movimiento, tal y como se conoce ahora, tiene sus inicios en la década de los 70 donde ya se comenzaba a utilizar en películas de animación. Pero es en la década de los 30 cuando se empezó a indagar sobre la captura de movimiento.

2.1.1 Historia de la captura de movimiento

Eadweard Muybridge, fotógrafo británico, consiguió diseccionar el movimiento de los animales y humanos mediante múltiples fotografías tomadas muy seguidas para captar instantes seguidos en el tiempo. Uno de los experimentos más conocidos de Muybridge es, sin duda, “El caballo en movimiento” mostrado en la Figura 2.1.^[13]

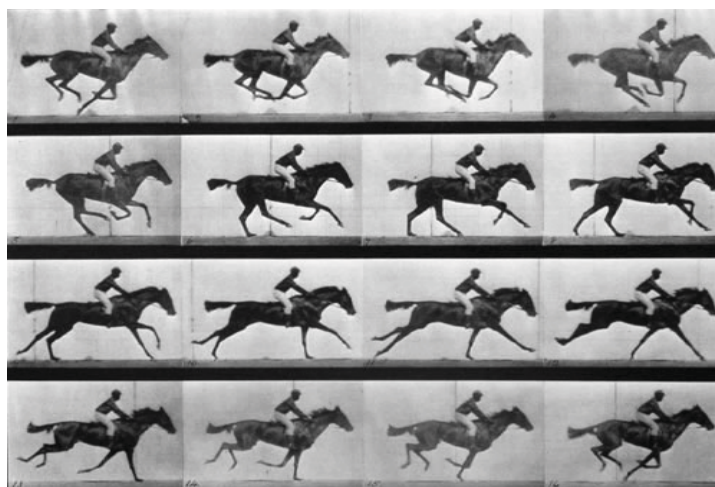


Figura 2.1. El caballo en movimiento de Eadweard Muybridge.

Con la aparición de la técnica de la rotoscopia se conseguía naturalizar los movimientos de los personajes animados. Los estudios Walt Disney Pictures utilizaron la rotoscopia en 1937 en “Blancanieves y los siete enanitos” para la animación de los personajes del príncipe y de Blancanieves. Esta técnica consiste en reemplazar los fotogramas de una grabación real por dibujos calcados sobre cada fotograma (Figura 2.2).^[14]



Figura 2.2. Ejemplo de uso de la técnica de la rotoscopia utilizada en la película de Blancanieves y los 7 enanitos de Disney.

En paralelo, los laboratorios de biomecánica comenzaban a utilizar ordenadores para analizar el movimiento humano. En la década de los 80, Tom Calvert, profesor de kinesiología y ciencias de la computación en la universidad Simon Fraser, Canadá, incorporó potenciómetros a un cuerpo y la salida la usó para mover personajes animados por ordenador para estudios coreográficos y asistencia clínica para pacientes con problemas de locomoción.

En 1988, deGraf/Wahrman desarrolla “Mike la cabeza parlante” de Silicon Graphics para mostrar las capacidades en tiempo real de sus nuevos equipos 4D (Figura 2.3). Mike era dirigido por un controlador especialmente construido para que permitiese controlar muchos parámetros de la cara del personaje, incluyendo la boca, ojos, expresión y la posición de la cabeza. El hardware de Silicon Graphics proporcionaba una interpolación en tiempo real entre las expresiones faciales y geometría de la cabeza del personaje y del artista. Mike fue mostrado ante el público en la conferencia de SIGGRAPH, grupo de interés en infografía o computación gráfica de la ACM, de ese año.

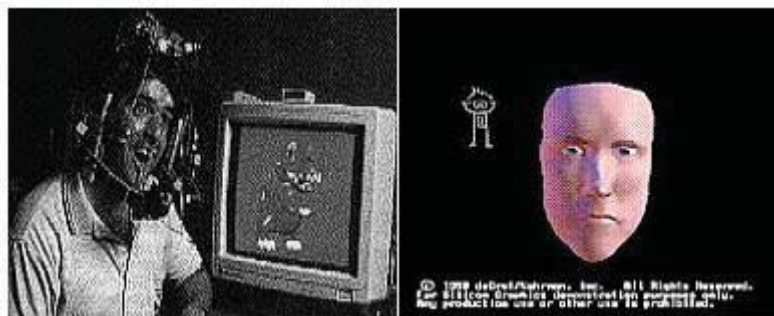


Figura 2.3. Mike la cabeza parlante, a la derecha. A la izquierda, el artista con el dispositivo que lo controlaba

Posteriormente, después de haber desarrollado “Mike la cabeza parlante” de deGraf/Wharman, Brad deGraf continuó investigando, en solitario, desarrollando un sistema de animación en tiempo real al que llamó “Alive!”. Creó un dispositivo especial para una mano con cinco pistones que se accionaban con los dedos del artista. Se utilizaba para controlar las expresiones faciales de una nave espacial, la cual, se usaba en una empresa para sus conferencias.^[15]

Con esto, deGraf pasó a formar parte de Colossal Pictures, donde implementó este sistema para animar a Moxy (Figura 2.4), un perro generado por ordenador que presentaba un programa para Cartoon Network. Moxy se manejaba en tiempo real en la publicidad y se renderizaba para la serie. Los movimientos del actor se capturaban con un sistema de captura electromagnético con sensores en manos, pies, torso y cabeza del mismo.



Figura 2.4. Moxy.

En la conferencia SIGGRAPH 93, Acclaim Entertainment aombró al público con una animación realista y bastante compleja de dos personajes hecha completamente con captura de movimiento. Acclaim había desarrollado en secreto un sistema de rastreo óptico

de alta definición, capaz de seguir 100 marcadores simultáneamente en tiempo real.^[16] Este sistema es usado principalmente para generar el movimiento de personajes de videojuegos. Se utilizó para la producción de muchos videojuegos de las diferentes compañías (Nintendo, Playstation...).

2.2 Tecnologías de captura de movimiento^[17]

Existen distintos tipos de tecnologías para realizar capturas de movimiento. A continuación, se realizará una breve descripción de las diferentes tecnologías.

a) Captura de movimiento electromecánica

Los sistemas de captura de movimiento electromecánicos (Figura 2.5) son aquellos sistemas en los que, en general, la captura de movimiento se realiza utilizando sensores mecánicos.

En el proceso de captura de movimientos, el actor o intérprete viste unos trajes especiales, adaptables al cuerpo humano. Los trajes son generalmente estructuras rígidas compuestas de barras metálicas o plásticas unidas mediante potenciómetros colocados en las principales articulaciones. El actor coloca la estructura en su cuerpo y mientras se mueve el traje se adapta a sus movimientos, y los potenciómetros recogen datos sobre el grado de apertura de las articulaciones.



Figura 2.5. Ejemplo de un sistema de captura de movimiento electromecánica.

Los potenciómetros constan de un elemento deslizante acoplado a una resistencia, que produce una variación de tensión que puede medirse para conocer el grado de apertura de la articulación a la que está acoplado, si bien los potenciómetros usados en captura de movimiento son mucho más complejos que los usados tradicionalmente en la industria electrónica. A veces, a estos sensores se les denomina sensores angulares digitales o analógicos. Los sensores angulares pueden enviar la información mediante cables, pero lo más frecuente es que lo hagan mediante radiofrecuencia.

Los sistemas de captura de movimiento electromecánicos tienen el problema, con respecto a otros sistemas de captura de movimientos, de ser incapaces de medir translaciones globales: son capaces de medir las posiciones relativas de los miembros, pero no el desplazamiento del actor en el escenario. Un sistema electromecánico completo tiene un precio comprendido entre 18.000 y 55.000 euros.

b) Captura de movimiento electromagnética

En los sistemas de captura de movimiento electromagnéticos (Figura 2.6) se dispone de una colección de sensores electromagnéticos que miden la relación espacial con un transmisor cercano. Se colocan en el cuerpo y se conectan a una unidad electrónica central, casi siempre mediante cables. Están constituidos por tres espiras ortogonales que miden el flujo magnético, determinando tanto la posición como la orientación del sensor.



Figura 2.6. Ejemplo de un sistema de captura de movimiento electromagnética.

Un sistema magnético típico consta de un transmisor, hasta 18 sensores, una unidad de control electrónica y un software propietario para el procesamiento. Un rastreador magnético de última generación puede tener hasta 90 sensores y es capaz de capturar hasta 144 muestras por segundo (este valor depende del número de sensores). El coste de estos sistemas varía de 4.000 a 11.000 euros.

c) Captura de movimiento óptica

Los sistemas ópticos utilizan los datos recogidos por sensores de imagen para inferir la posición de un elemento en el espacio, utilizando una o más cámaras sincronizadas para proporcionar proyecciones simultáneas. Lo habitual es que los datos se recojan utilizando indicadores, o markers, pegados al actor, pero los sistemas más recientes permiten recoger datos fiables rastreando superficies del sujeto identificadas dinámicamente.

Los sistemas ópticos más habituales se basan en un único ordenador que recibe la entrada de varias cámaras digitales CCD (Charge-Coupled Device). El número de cámaras se comprende habitualmente entre 4 y 32. Aunque sería suficiente con dos cámaras para obtener la posición de cualquier indicador.

- **Captura de movimiento mediante indicadores pasivos**

Los indicadores pasivos (Figura 2.7) están recubiertos de un material reflectante y se pegan al actor en puntos estratégicos. La luz que reflejan se origina cerca de las cámaras, y es recogida por éstas.

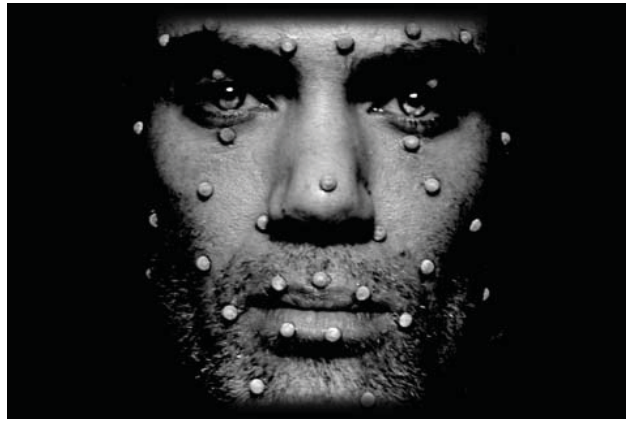


Figura 2.7. Ejemplo de un sistema de captura de movimiento óptica mediante indicadores pasivos.

Los sistemas ópticos mediante indicadores pasivos no necesitan que el actor se coloque dispositivos electrónicos en el cuerpo, sino un gran número de bolas de goma recubiertas de material reflectante. Este tipo de sistemas pueden capturar un gran número de marcadores a frecuencias del orden de hasta 2000 fotogramas/segundo, si bien hay que compensar la velocidad de captura con la resolución. Existen sistemas de 4 megapíxeles a 360 hercios entorno a los 74.000 euros, y sistemas de 0,3 megapíxeles a 120 hercios alrededor de los 18.000 euros.

- **Captura de movimiento mediante indicadores activos**

En este sistema de captura (Figura 2.8), los indicadores emiten su propia luz mediante LEDs, lo cual se consigue aumentar la distancia a la que se puede desplazar el sujeto. La posición de los indicadores se determina iluminando un indicador en cada instante de tiempo a una frecuencia muy alta, o bien varios indicadores de cada vez, con procesamiento adicional para calcular la identidad de cada indicador a partir de su posición relativa. Para ello, los indicadores han de estar sincronizados con todas las cámaras para iluminarse en una sola captura.



Figura 2.8. Ejemplo de un sistema de captura de movimiento óptica mediante indicadores activos.

- **Captura de movimiento mediante indicadores activos modulados en el tiempo**

Se trata de una mejora con respecto a los indicadores activos, en la cual los marcadores no se iluminan uno de cada vez, sino que se iluminan muchos a la vez mediante luz estroboscópica, determinándose la identidad de cada indicador mediante la frecuencia de destello. De esta forma se consiguen frecuencias de captura mayores que con los sistemas activos estándar, con el inconveniente de aumentar la carga computacional, si bien esto en la actualidad no supone un problema. Existen sistemas de captura de movimiento compuestos por ocho cámaras de 12 megapíxeles capaces de capturar hasta 480 fotogramas por segundo a precios inferiores a 37.000 euros.

- **Mediante indicadores semi-pasivos imperceptibles (Semi-passive Imperceptible Marker)**

Los sistemas de captura de movimiento mediante indicadores semi-pasivos imperceptibles se caracterizan porque son los propios indicadores los que detectan su propia posición y orientación.

Estos sistemas permiten un número ilimitado de etiquetas indicadoras y, al no utilizarse cámaras de alta definición, se reduce considerablemente el tráfico de datos generado. Cada indicador procesa su posición y ésta no tiene que calcularse a partir de imágenes. Estos sistemas son ideales para la captura de movimientos en tiempo real.

- **Sin marcadores**

Son sistemas en los que no se requieren trajes especiales para el seguimiento de los movimientos de los actores. Se basan en una serie de algoritmos que analizan distintas fuentes de entrada de imágenes identificando formas humanas y descomponiéndolas en segmentos para realizar el seguimiento de sus movimientos. Estos sistemas trabajan bien con movimientos generales pero suelen tener dificultades con movimientos sutiles, como los de los dedos de la mano.

d) Captura mediante fibra óptica

Los primeros sistemas de este estilo son los guantes de fibra óptica, pero en la actualidad se intenta aplicar esta técnica a la captura de movimientos del cuerpo entero. Los guantes de fibra óptica están constituidos por un conjunto de fibras ópticas que, al doblarse atenúan la luz transmitida, permitiendo calcular la posición de los dedos de la mano.

Para la captura del cuerpo entero, se fijan sobre las distintas partes del cuerpo sensores flexibles de fibra óptica que miden las rotaciones de las articulaciones. Al igual que los sistemas electromecánicos, no se mide la posición del actor en el escenario.

e) Captura mediante ultrasonidos

En estos sistemas, en general, se utilizan emisores que generan pulsos ultrasónicos (imperceptibles por los seres humanos) que son capturados por uno o varios receptores situados en posiciones conocidas, permitiendo averiguar la posición del emisor en el espacio, e incluso su orientación en algunos casos.

f) Captura mediante sistemas inerciales

Los sistemas inerciales utilizan unos pequeños sensores (acelerómetros y giroscopios) que recogen información sobre la aceleración y la velocidad angular del sensor. Conociendo la posición y la velocidad angular inicial e integrando las informaciones que recogen los sensores, es posible determinar la posición, eje de giro y velocidad angular de cualquier sensor. Este tipo de sistemas de captura de movimiento no utiliza mecanismos externos como cámaras; y como en el caso de los sistemas ópticos, cuantos más sensores se utilicen, más real es el movimiento reproducido.

2.3 Captura de movimiento con Kinect^[2]

Kinect es un controlador de videojuegos desarrollado por Microsoft para la videoconsola, de la misma compañía, Xbox 360. Con Kinect se puede interactuar con la consola mediante gestos o por comandos de voz sin necesidad de un controlador físico tradicional.

Kinect cuenta, a nivel hardware, con una cámara RGB, un sensor de profundidad, un micrófono multirray y un procesador que ejecuta un software patentado para la captura de movimiento de todo el cuerpo en 3D en tiempo real. El micrófono multirray permite localizar la fuente acústica y suprimir el ruido ambiental. La cámara cuenta con una resolución 640x480 píxeles, y envía datos con una frecuencia de refresco de 30 fps, con lo que algunos movimientos tendrán algunos frames de retraso.

El sensor de profundidad es un proyector de infrarrojos con un sensor de tipo CMOS monocromo que permite a Kinect trabajar bajo cualquier condición de luz ambiental. El rango de detección de la profundidad del sensor es ajustable gracias al software interno de Kinect que es capaz de calibrar automáticamente el sensor.



Figura 2.9. Imagen del Kinect de Microsoft.

El proceso de la captura de movimiento con Kinect, comienza con los sensores de profundidad que emiten luz infrarroja, la cual rebota en la persona y vuelve a los sensores. A continuación, se calcula el tiempo que tarda esta luz en salir y en volver. De esta manera el dispositivo conoce la profundidad, es decir, donde está parada la persona. El software interno se encarga, a través de la cámara RGB, en llevar a cabo una captura de movimiento óptica en tiempo real (y sin sensores). La captura en tiempo real se refleja en los personajes de manera instantánea.

En cuanto al uso de este dispositivo en un ordenador, Microsoft liberó los drivers para facilitar su aplicación en entornos Windows, popularizando así su uso en múltiples aplicaciones, tales como la captura de movimiento. Además, algunas compañías dedicadas especialmente a la captura de movimiento, han desarrollado software compatibles con la Kinect.

Para este Trabajo Fin de Grado se ha decidido utilizar el software NI-MATE,^[18] ya que está desarrollado especialmente para la captura de movimiento. Además, contiene los controladores para usar la Kinect de Microsoft.

Capítulo 3.

Software de diseño 3D

Para la captura de movimiento, además de la tecnología, se precisa de la ayuda de los softwares de diseño 3D. Por ello, en este capítulo se hará un pequeño análisis de los softwares de diseño 3D que hay en el mercado. A su vez, se hará una descripción del software escogido para este trabajo, el Blender.

3.1 Software para diseño 3D

Hoy en día, en todas partes puede verse el fruto del uso de programas de diseño 3D (televisión, publicidad...). Además, cada vez más, salen a la luz nuevos softwares con una amplia variedad de opciones y herramientas para este fin. Se pueden diferenciar en dos tipos: Software comercial (requiere un desembolso), software libre (es gratuito).

3.1.1 Software comercial

Actualmente, existe una amplia variedad de software para el diseño y animación 3D en el mercado. A continuación se describen las características de los más usados.

- **Autodesk 3Ds Max**^[19]

Es un programa de modelado y animación 3D profesional desarrollado por Autodesk. A menudo se utiliza en arquitectura, ingeniería, cine, TV, juegos y proyectos publicitarios. Además, permite crear personajes animados a través del modelado y animación 3D. Ofrece una variedad de características de gran alcance y flujos de trabajo que permiten a los usuarios crear cualquier lugar desde simples a complejos diseños arquitectónicos, virtualizaciones, personajes de videojuegos, películas de animación y fotorrealistas efectos de cine en 3D (Figura 3.1).

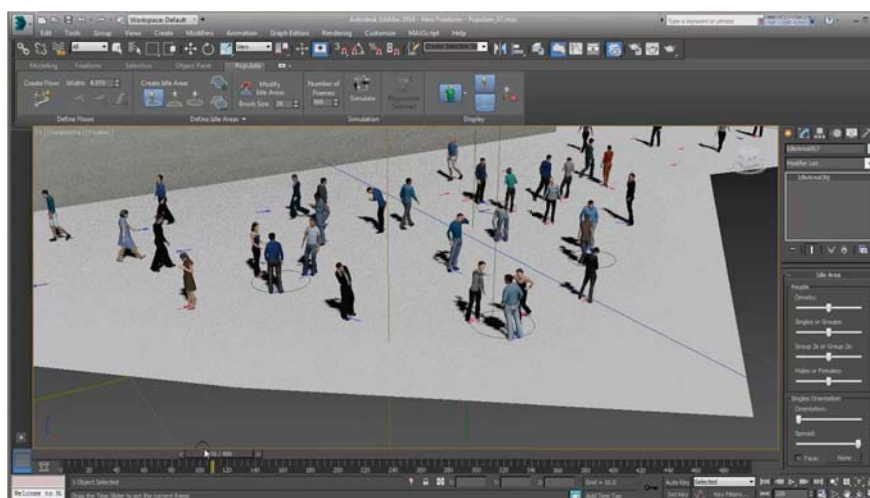


Figura 3.1. Muestra del software 3ds Max.

Sin embargo, este software está hecho únicamente para los sistemas operativos de Microsoft. La última versión está disponible para Windows 7 y Windows 8. Esto es una clara desventaja, ya que se debe disponer de un ordenador con dicho sistema operativo para poder trabajar con él. Además, se obliga al usuario a pagar una licencia para poder usarlo. La última versión ofrece una prueba gratuita pero está muy limitada, ya que sólo permite usarla durante 30 días.

- **Autodesk Maya**^[20]

Maya es un programa informático dedicado al desarrollo de gráficos 3D por ordenador, efectos especiales y animación. Además, Maya es el único software de 3D acreditado con un Oscar gracias al enorme impacto que ha tenido en la industria cinematográfica como herramienta de efectos visuales, con un uso muy extendido debido a su gran capacidad de ampliación y personalización.

El programa posee diversas herramientas para modelado, animación, renderización, simulación de ropa y cabello, dinámicas (simulación de fluidos), etc. Maya trabaja con cualquier tipo de superficie NURBS, polígonos y subdivisión de superficies. Ya que, gracias al código que forma el núcleo de Maya, se pueden crear scripts y personalizar el paquete del software (Figura 3.2).



Figura 3.2. Muestra del software Autodesk Maya.

Maya se puede utilizar con Mac OSX, con Windows y con Linux; es multiplataforma. Pero, al igual que el software de la misma compañía 3ds Max, requiere de una licencia. También ofrece una licencia gratuita que te permite usarlo durante tres años, pero a partir de esos tres años se deberá contratar dicha licencia.

- **Cinema 4D**^[21]

Cinema 4D (Figura 3.3), es un software de creación de gráficos y animación 3D desarrollado originariamente para Commodore Amiga por la compañía alemana Maxon, y portado posteriormente a plataformas Windows y Macintosh (OS 9 y OS X). Tiene una versión gratis para estudiantes. Además, no requiere de una licencia anual como los anteriores softwares de los que se ha hablado.



Figura 3.3. Muestra del software Cinema 4D

Permite modelado (primitivas, splines, polígonos), texturización y animación. Sus principales virtudes son una muy alta velocidad de renderización, una interfaz altamente personalizable y flexible, y una curva de aprendizaje (comparado con otros programas de prestaciones profesionales similares) muy vertical; en poco tiempo se aprende mucho. Una de las características más destacadas de Cinema 4D es la modularidad.

3.1.2 Software libre

Cada vez son más los software de diseño 3D libres y de código abierto. A continuación se comentan las características de los más usados, entre ellos, el Blender.

- **Google Sketchup**^[22]

Es un programa de diseño gráfico y modelado 3D basado en caras. Su principal característica es la de poder realizar diseños complejos en 3D de forma extremadamente sencilla. El programa incluye entre sus recursos un tutorial en vídeo para ir aprendiendo paso a paso cómo se puede ir diseñando y modelando el propio ambiente. Permite conceptualizar y modelar imágenes en 3D de edificios, coches, personas y cualquier objeto o artículo que imagine el diseñador o dibujante. Además el programa incluye una galería de objetos, texturas e imágenes listas para descargar (Figura 3.4).

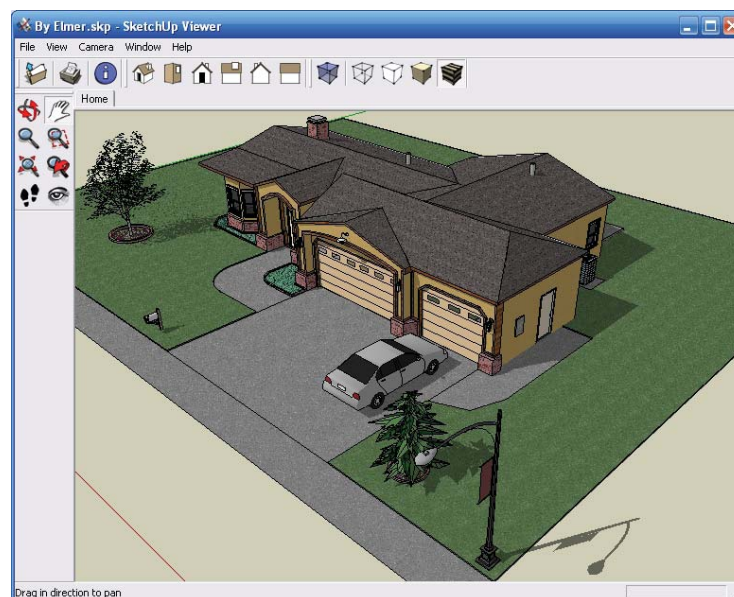


Figura 3.4. Muestra del software Google SketchUp.

Google sketchup está muy limitado en cuanto a animación, ya que para lo único que sirve en ese campo es para crear estructuras y objetos, pero no para animarlos.

- **Wings 3D**^[23]

Wings 3D (Figura 3.5) es un programa de modelado 3D libre. Está disponible para la mayoría de plataformas, incluyendo Windows, Linux y Mac.



Figura 3.5. Muestra del software Wings 3D.

Está diseñado para modelar y texturizar elementos formados con un pequeño número de polígonos. Cuando se compara con otros programas de 3D (como puede ser Blender) se observan las diferencias existentes sobre todo en la interfaz gráfica de usuario, la cual aporta una mayor flexibilidad a cambio de ciertas limitaciones en algunas áreas. Por ejemplo, Wings 3D no puede manejar animaciones, sólo trae un renderizador de OpenGL y muchas opciones pueden llegar a desactivarse si el elemento poligonal es demasiado complejo. Wings 3D es de muy fácil manejo y posee un sistema de iconos muy intuitivo.

- **K-3D**^[24]

K-3D (Figura 3.6) es un software de modelado 3D y animación libre. Es una herramienta versátil ya que posee una amplia gama de plugins totalmente combinables. Es muy intuitivo y potente. Posee un sistema avanzado de “undo/redo” (deshacer /rehacer) que garantiza que siempre se pueda ir hacia atrás y hacia adelante, incluso hasta donde otros programas pierden la capacidad de rehacer. El innovador sistema tutorial interactivo de K-3D dará a conocer el uso básico del programa. K-3D está escrito en C++, y está disponible para Linux, MacOS X, Microsoft Windows entre otros.

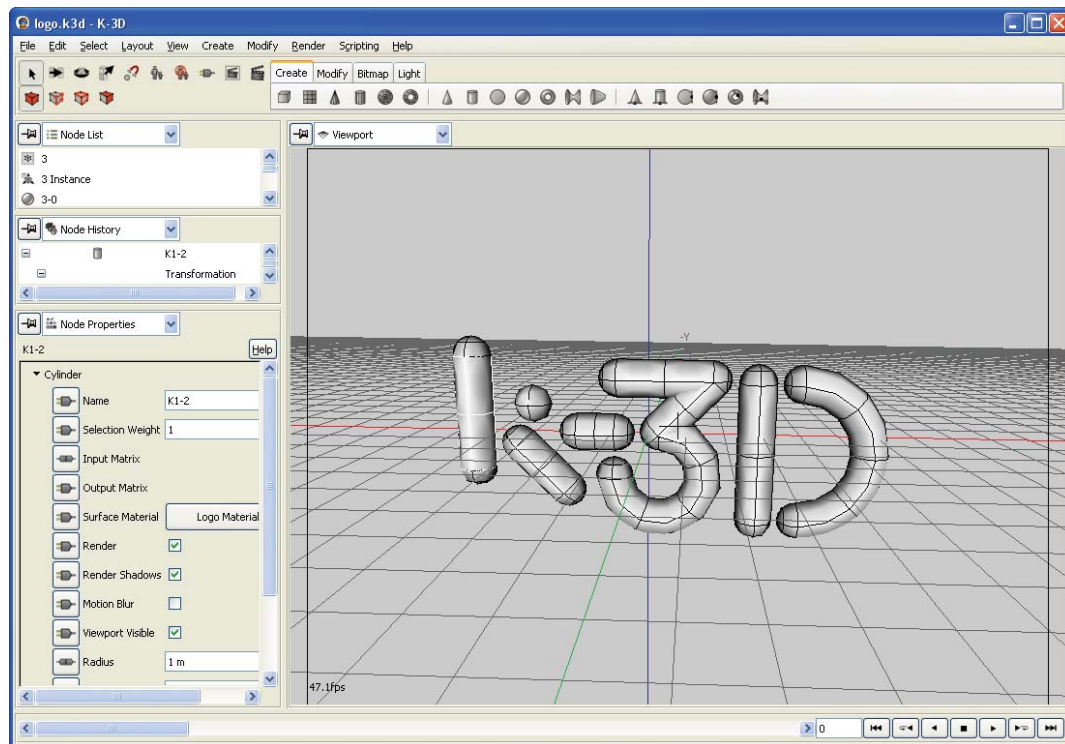


Figura 3.6. Muestra del software K-3D.

- **Blender**^[8]

Blender (Figura 3.7), es un programa informático multiplataforma, dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. Además, es un software de composición digital ya que tiene la capacidad de edición de vídeos, escultura y pintura digital. En Blender, también, se pueden desarrollar videojuegos ya que posee un motor de juegos interno. Actualmente es compatible con todas las versiones de Windows, Mac OS X, GNU/Linux, Solaris, FreeBSD e IRIX.



Figura 3.7. Muestra del software Blender.

Una de las características principales del software es su tamaño. Blender tiene un tamaño de origen realmente pequeño en comparación con otros paquetes de 3D. Esto es una ventaja considerable, ya que, al ser multiplataforma, libre y de pequeño tamaño, en cualquier ordenador con conexión a internet se podrá descargar y continuar con los proyectos.

Como el fin de este Trabajo de Fin de Grado es encontrar una solución de bajo coste de captura de movimiento, se ha escogido el software Blender, ya que de entre los nombrados anteriormente, es el que cumple con lo necesario para ejecutar este trabajo. Blender es software libre y para la realización de la captura de movimiento existen una serie de plugins, los cuales se expondrán más adelante.

A continuación, en el siguiente apartado, se describirán un poco más en profundidad sus características.

3.2 Blender

Como ya se comentó anteriormente, Blender es un software de modelado y animación 3D. Por medio de su arquitectura abierta, ofrece interoperabilidad entre plataformas, extensibilidad, ocupa un espacio increíblemente pequeño en disco duro, y un flujo de trabajo altamente integrado. Blender es una de las aplicaciones de Código Abierto de gráficos 3D más populares del mundo. A continuación, se expondrán algunas de sus características principales.

3.2.1 Características principales de Blender

Como ya se comentó, es un software multiplataforma, libre y gratuito. Posee una alta capacidad para una gran variedad de primitivas geométricas, como curvas, mallas poligonales, NURBS, metaballs. En la Figura 3.8 se puede ver un ejemplo de un modelo hecho con primitivas renderizado en Blender.

Además, posee unas características interactivas para juegos como detección de colisiones, recreaciones dinámicas y lógicas. Herramientas de animación, tales como, la posibilidad de deformaciones en las mallas. Contiene herramientas básicas de edición de audio y sincronización de video. Además, Blender acepta formatos gráficos como TGA, JPG, Iris, SGI o TIFF. También tiene la utilidad de modificadores apilables, para la aplicación de transformación no destructiva sobre mallas, además de un sistema de partículas estáticas para simular cabellos y pelajes.

Con Blender, se pueden recrear varias simulaciones físicas, como: sólidos rígidos, sólidos deformables, fluidos y hasta humo de una forma muy sencilla e intuitiva.



Figura 3.8. Ejemplo de modelo renderizado en Blender.

Blender posee la capacidad de hacer “match moving”. El match moving es una técnica de efectos visuales que permite insertar gráficos creados por ordenador en un video con la posición correcta, escala, orientación y movimiento en relación a los objetos presentes en la toma.

3.2.2 Blender y la captura de movimiento

Blender permite trabajar con mocap de dos maneras: con datos externos (BVH) o hacer la captura en tiempo real.

a) Datos Externos (BVH)

BVH (Biovision Hierarchy) es un tipo de formato de archivo que contiene información jerárquica de un esqueleto y el movimiento que realiza cada hueso. Blender da la posibilidad de importar este tipo de archivos. Al importarlos genera un esqueleto, el cual realiza el movimiento que contiene. Para que un modelo cualquiera se mueva, hay que establecer un seguimiento hueso a hueso del modelo con el generado con los datos externos.

b) Captura en tiempo real

Blender no dispone de un módulo interno específico que permita hacer, desde el mismo software, captura de movimiento en tiempo real. Por ello, hay que utilizar un software externo con el que poder realizarlas.

Blender da la posibilidad de añadirle lo que se conoce como “addons” (plugins). Existen una amplia gama de addons, al ser de código abierto, hay mucha variedad en la red. Según el software de captura de movimiento que se desee usar, se podrá elegir entre uno u otro.

Existe un software llamado Ni-mate, que es el que se ha usado para este Trabajo de Fin de Grado, para el cual se ha realizado un pequeño tutorial para su uso, anexo al final de este trabajo, dispone de un addon para implementarlo en Blender.

Capítulo 4.

Desarrollo del tutorial

Una vez conocida la tecnología que se usará para la realización de la captura de movimiento y el software, en este capítulo, se describirá el proceso que se ha seguido para el desarrollo del tutorial, anexo a este Trabajo de Fin de Grado.

4.1 Objetivos

El objetivo del desarrollo del tutorial de este Trabajo de Fin de Grado es proveer, tanto al alumno como al docente, de los conocimientos básicos acerca del MOCAP.

El mundo de la animación se actualiza a pasos de gigante. Los animadores que trabajan actualmente con ello, se han dado cuenta que están ante un público exigente, el cual requiere más y más realismo tanto en películas de animación como en juegos. Dicho realismo, se consigue con horas y horas de trabajo. En especial, para animar manualmente personajes de películas tan reales, los animadores tendrían que trabajar años. Por ello, en la actualidad, se emplean técnicas como la del MOCAP, o captura de movimiento.

En los objetivos y competencias de la mención de Sonido e Imagen del Grado en Ingeniería en Tecnologías de la Telecomunicación, está establecido, que, de entre las competencias del graduado, con dicha mención, esté la de “la capacidad para crear, codificar, gestionar difundir y distribuir contenidos multimedia, atendiendo a criterios de usabilidad y accesibilidad de los servicios audiovisuales, de difusión e interactivos”. En la asignatura, de la mención antes nombrada, “Postproducción Digital y Animación” se estudia todo lo relacionado con las postproducción digital, además de animación 2D y 3D. Por eso, disponer de una herramienta que proporcione una visión general de la captura de movimiento será una fuente útil para el proceso de aprendizaje. Con esto el alumno aumentará sus conocimientos en el mundo de la animación actual.

4.2 Diseño del tutorial

El tutorial está constituido por un documento en formato PDF, anexo a este Trabajo de Fin de Grado, además de los archivos necesarios para la realización del mismo y un video demostrativo.

En el documento PDF se ha intentado que sea lo más gráfico posible. Para ello se han explicado todos los pasos, no sólo con texto, ya que también se han añadido imágenes para reforzar la comprensión del mismo.

Todos los archivos que se usarán durante la realización del tutorial se encuentran en una carpeta almacenada en el CD adjunto a este Trabajo de Fin de Grado. Se han ordenado por carpetas y se le ha puesto a cada archivo nombres de fácil identificación, además de referenciarlos de forma clara en el texto, para que el lector acceda a ellos sin problema alguno.

En cuanto al vídeo demostrativo, se han grabado diferentes movimientos y en la edición se le ha añadido una canción para que sea más amena la visualización. En el vídeo se podrá ver los diferentes modelos usados en el tutorial, moviéndose al mismo tiempo que una persona normal, es decir, realizando MOCAP.

4.3 Material utilizado

Durante el desarrollo del tutorial se han necesitado una serie de recursos hardware y software que se exponen a continuación.

4.3.1 Software

Los softwares utilizados para el desarrollo del tutorial, a parte de los usados para la realización del MOCAP, son:

- El Microsoft Word, para la redacción del documento.
- El Fraps, para realizar capturas de pantalla completa.
- El Recortes, para realizar capturas de pequeños detalles de pantalla.
- Adobe Photoshop, para retocar algunas capturas.
- Adobe Premiere, para la edición del vídeo demostrativo.
- El Newtek, para enviar lo que se muestra por la pantalla a un mezclador para el vídeo demostrativo.

4.3.2 Hardware

- Un dispositivo Kinect de Microsoft para la captura de movimiento.
- Estación de trabajo basada en Intel Pentium Dual Core, 4 GB RAM. Monitor 21” TFT LCD.
- Cámara profesional de vídeo Sony HVR-Z1E.
- Set de estudio con ciclorama e iluminación profesional.
- Ordenador personal.

4.4 Proceso de realización del tutorial

Para el desarrollo del tutorial, se han seguido 4 fases que se describen a continuación.

4.4.1 Fase 1. Estructura

En esta fase, se ha establecido una estructura para la realización del documento del tutorial. Para ello se ha desglosado, la información recabada, y se ha planteado una distribución a lo largo de las 3 partes del tutorial que quede de manera continua para el lector.

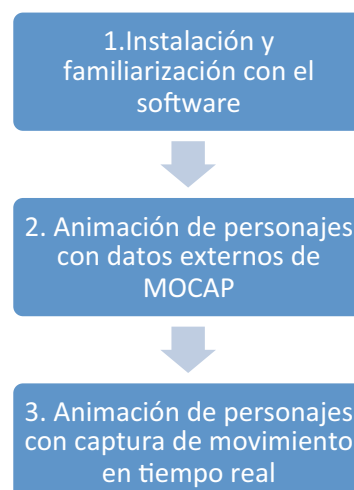


Figura 4.1. Partes en las que se divide el tutorial.

En la primera parte del tutorial se describe paso a paso el proceso de descarga e instalación de los softwares que se usarán para realizar la captura de movimiento. Seguidamente, se realiza una guía básica de las herramientas que ofrece el software Blender, que serán necesarias para la realización del tutorial.

Una vez se ha familiarizado el alumno con el entorno de Blender, en la parte 2, se muestra el procedimiento, para la realización de captura de movimiento mediante datos externos. Con esto, se pretende que el lector aprenda cuáles son las fases por las que se debe pasar para poder conseguir que cualquier modelo 3D se mueva mediante la técnica de MOCAP.

Finalmente, ya comprendido el funcionamiento de la técnica de captura de movimiento, se muestra el proceso para la realización de una captura de movimiento en tiempo real, es decir, conseguir poner en movimiento un modelo 3D al mismo tiempo que el individuo que se está capturando. Una vez realizadas todas las partes, el lector será capaz de trabajar con esta técnica para añadir movimientos reales a sus proyectos de animación.

4.4.2 Fase 2. Formato

En esta fase, se ha generado un guión, el cual es el texto redactado en el tutorial. Se ha intentado realizar un texto sencillo de entender, con un tamaño 12 y un tipo de letra “Times New Roman” para que al lector le sea más cómoda la lectura.

Como ya se ha comentado, el documento está dividido en 3 partes. Cada parte está claramente señalizada para que el lector sepa siempre en cuál se encuentra. A su vez, cada parte está dividida en sub-apartados. De esta manera, la información queda desglosada con lo que se puede seguir fácilmente. Cada sub-apartado contiene un texto explicativo con lo que se va a realizar y está reforzado con imágenes para la mayor comprensión del mismo.

En la siguiente figura, se señala lo antes comentado. El número 1, en la figura, se corresponde con la división de sub-apartados. El número 2, señala el encabezado, el cual indica en que parte del tutorial se encuentra. El 3, es un ejemplo de una de las imágenes añadidas para el refuerzo del texto. El número 4 indica el número de página en la que se encuentra el lector en cada momento.

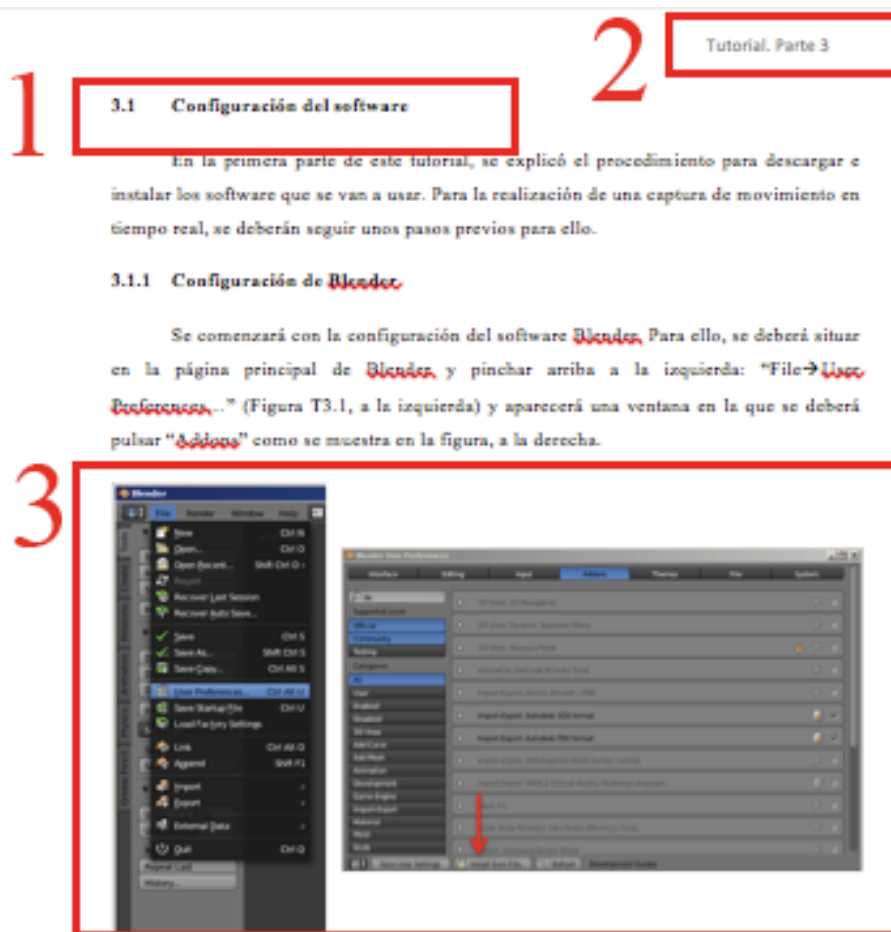


Figura T3.1. A la izquierda, como llegar a "User Preferences...". A la derecha, la ventana "Add-ons".

Una vez allí, en la parte de abajo de dicha ventana, se deberá pulsar "Install From File..." que aparece señalado con una flecha en la Figura anterior (Figura T3.1, a la derecha). A continuación, deberá buscar el archivo "animation_decode_nimate_tools.py" que está incluido en la carpeta "configuración → Add-on Blender" de los archivos adjuntos a este tutorial. Al seleccionarlo, se deberá pulsar "Install From File..." situado arriba a la derecha.

TFG. Solución de bajo coste de captura de movimiento basada en Kinect

4 31

Figura 4.2. Muestra del formato empleado en el tutorial.

4.4.3 Fase 3. Producción de contenidos

Se han realizado las capturas de pantalla necesarias para la mejor comprensión del texto y se han usado diferentes herramientas de captura y retoque digital, entre ellas, Fraps para realizar capturas de pantalla, y Adobe Photoshop CS6 para el retoque de las capturas realizadas.

En el tutorial, como se explicó en el apartado anterior, se han añadido imágenes para reforzar la comprensión del texto. Para la obtención de estas imágenes, se ha seguido un proceso que se expone a continuación:

a) Captura de la imagen

En esta fase del proceso se han utilizado el software Fraps y la herramienta Recortes que proporciona el sistema operativo Windows 7.

Fraps es un software para realizar capturas de vídeo y de pantalla. Se ejecuta en segundo plano y no consume muchos recursos. Para la realización del tutorial, se ha usado para hacer capturas de la interfaz de Blender a pantalla completa, mientras se trabajaba en este, ya que Fraps tiene una opción de realizar capturas de pantalla cada cierto tiempo, antes configurado. En este caso se realizaban capturas cada 10 segundos, sobre todo, cuando se quería realizar capturas de un proceso.^[27] En la Figura 5.3 se muestra la interfaz del programa.

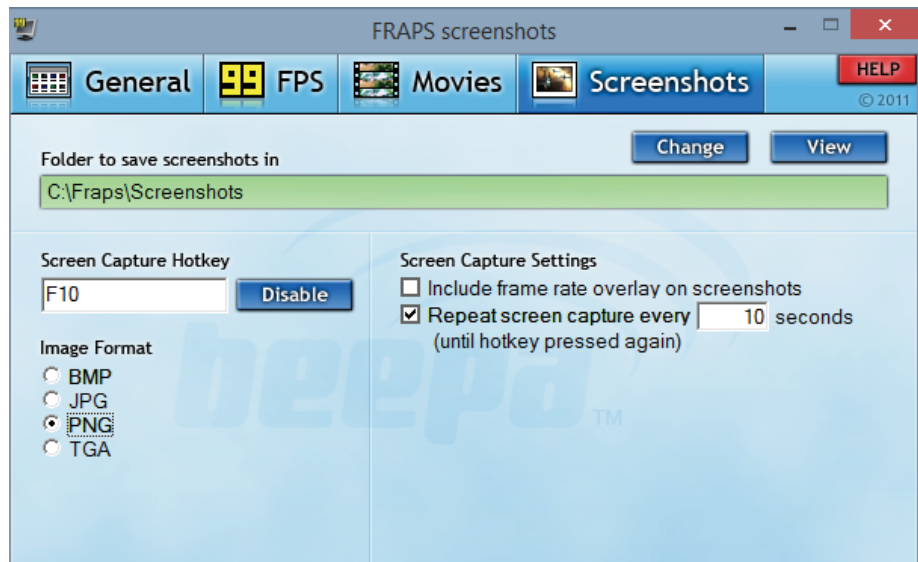


Figura 4.3. Muestra de la interfaz del software Fraps.

En cuanto a la herramienta Recortes, es una herramienta que nos proporciona tanto, Windows 7 como Windows 8. Se utiliza para hacer capturas de pantalla, pero, además de capturas a pantalla completa, pueden realizarse de sectores específicos de la pantalla. En la siguiente figura se muestra su interfaz.

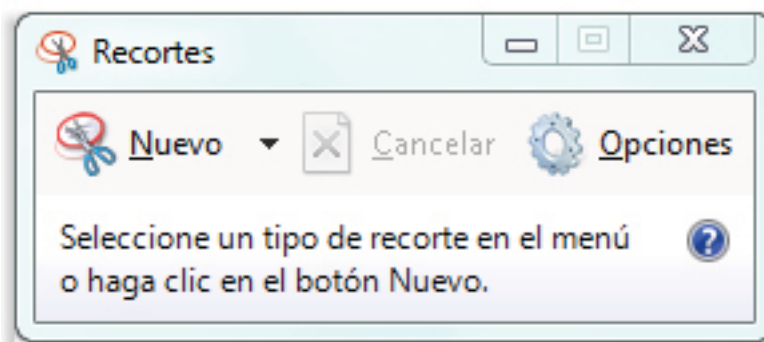


Figura 4.4. Muestra de la herramienta Recortes de Windows 7.

b) Retoque fotográfico de las capturas realizadas

La mayoría de las capturas realizadas estaban oscuras y casi no se apreciaban los detalles o necesitaban de algún tipo de señalización como indicación para el lector. Debido a esto, se ha precisado del uso de software de retoque fotográfico. En este trabajo se ha utilizado el software Adobe Photoshop, un software especialmente dedicado a retoque y edición fotográfica.^[25]

Para este trabajo se han retocado 66 de 77 capturas. Todas estas capturas han seguido el mismo proceso de retoque y edición. El proceso ha sido realizar ajustes de brillo y contraste, ya que , al estar oscuras, impedía verlas con normalidad. En las siguientes figuras se muestra un ejemplo del antes y después del ajuste de brillo.

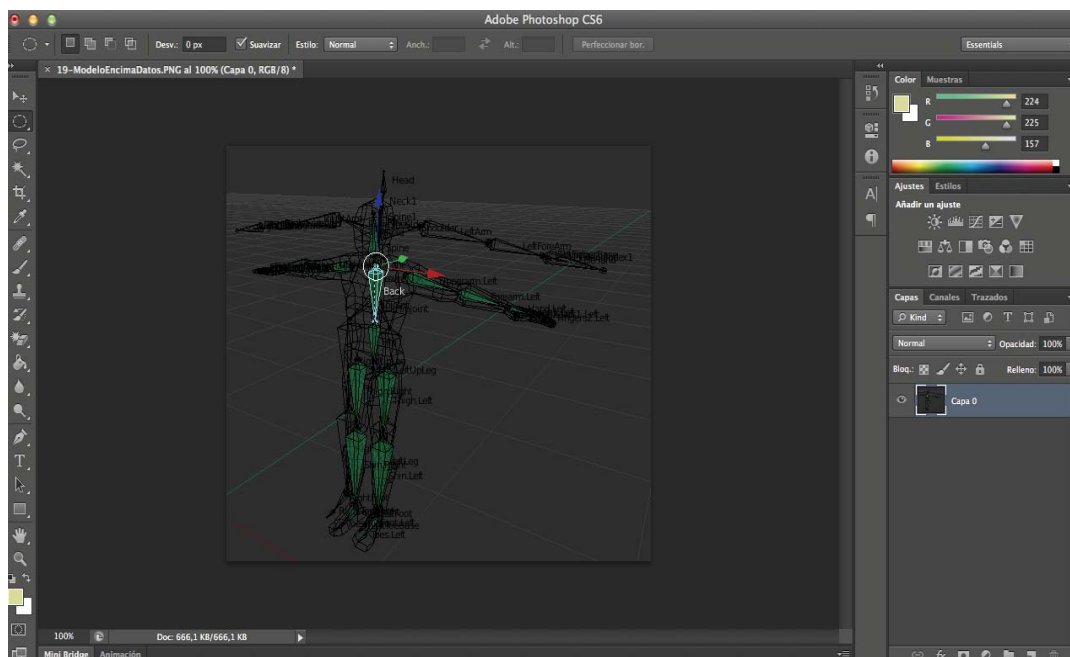


Figura 4.5. Ejemplo de captura sin retoque fotográfico.

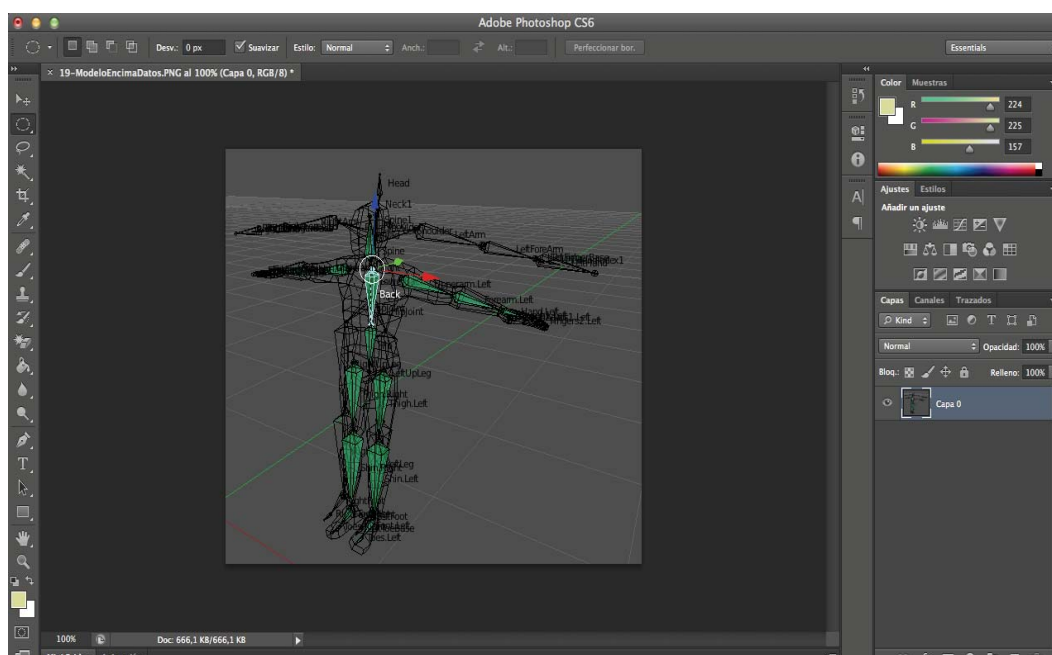


Figura 4.6. Ejemplo de la captura ya retocada.

Además de los ajustes de brillo que se comentó anteriormente, algunas imágenes precisaban de algún tipo de señalización indicativa. Esta tarea se ha realizado, también, con el Adobe Photoshop. A continuación un ejemplo del antes y después de señalizarla.

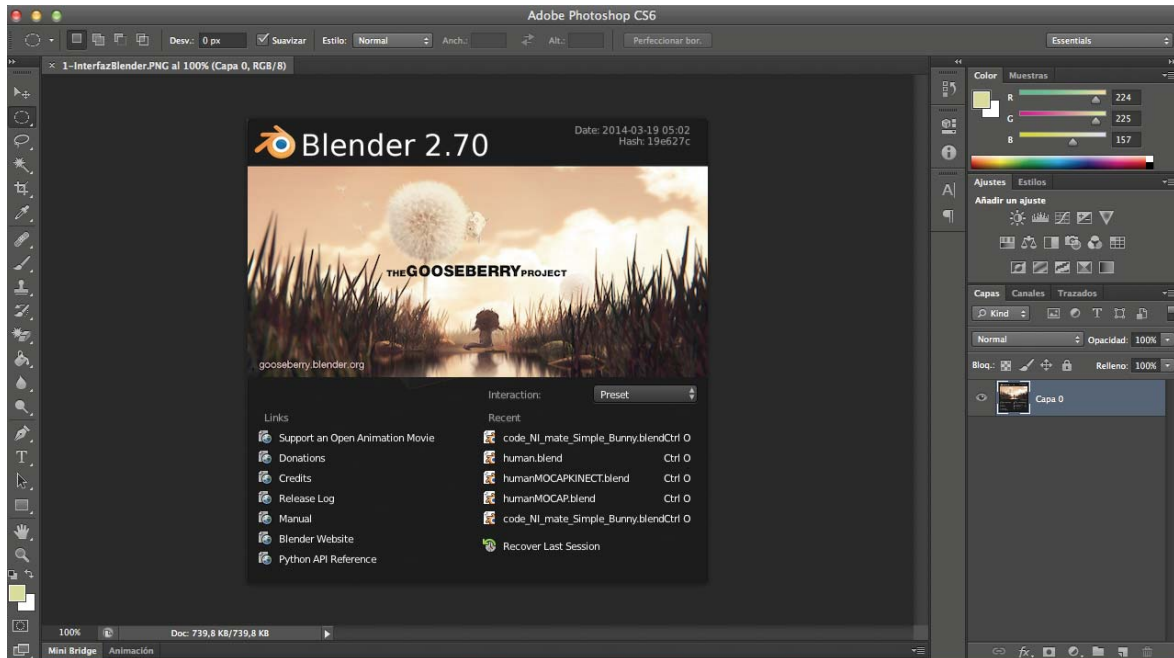


Figura 4.7. Ejemplo de una captura sin señalización.

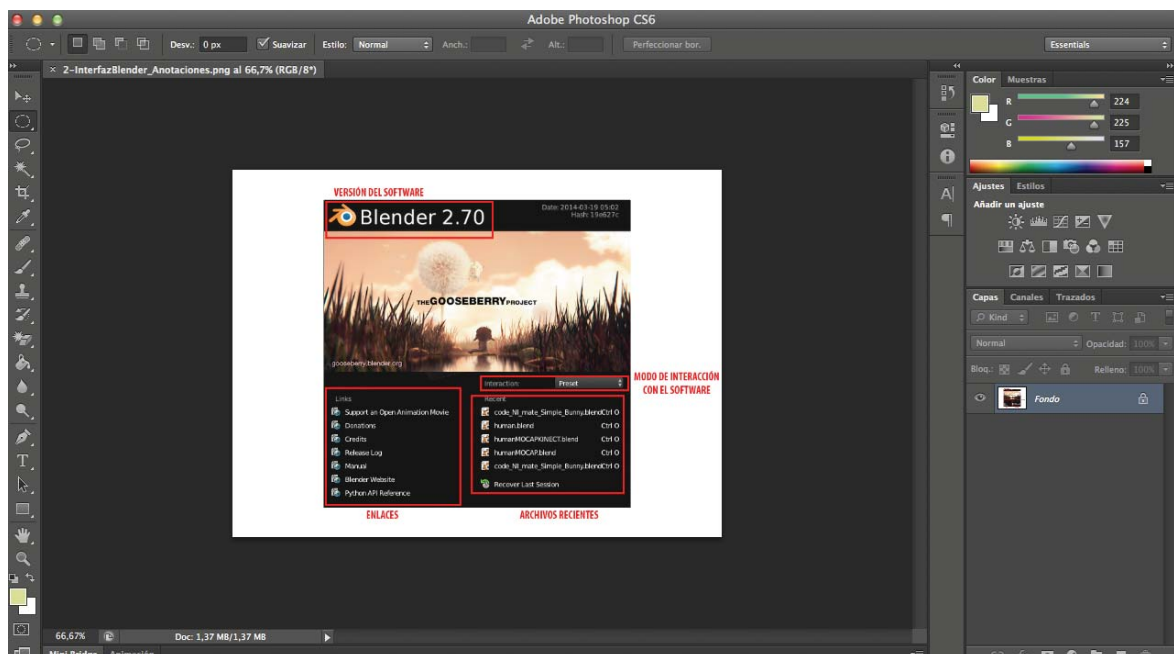


Figura 4.8. Ejemplo de una captura con señalización.

4.4.4 Fase 4. Material de apoyo

Para un mejor seguimiento del tutorial, en el CD, adjunto a este Trabajo de Fin de Grado, se han añadido una serie de archivos y un vídeo demostrativo que se describen en los siguientes apartados.

- **Archivos del tutorial**

En el CD que se adjunta con es trabajo, se incluye un archivo comprimido en el que se ha depositado una carpeta con los archivos necesarios para seguir el tutorial. Esta carpeta incluye tres carpetas: “Configuración”, “DatosBVH” y “Modelos” (Figura 4.9).

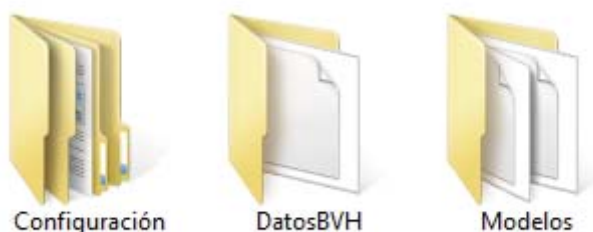


Figura 4.9. Muestra de las carpetas que se adjuntan con el tutorial.

En la carpeta “Configuración” se encuentran los archivos necesarios para realizar la implementación del software Delicode Ni-Mate con el software Blender.

En la carpeta “DatosBVH” se ha incluido un archivo “.bvh” para realizar la parte de captura de movimiento con datos externos. Además se ha generado un documento “.txt” donde aparecen páginas web donde poder conseguir de manera gratuita más archivos externos de mocap.

Por último, en la carpeta “Modelos”, se encuentran todos los proyectos de Blender con los modelos que se necesitan para realizar el tutorial. Para cada modelo, se ha comprobado, como se muestra en la Figura 4.10, que tuviese lo necesario para la realización de la captura de movimiento, por ejemplo, que contenga esqueleto o que la malla estuviese ligada al esqueleto.

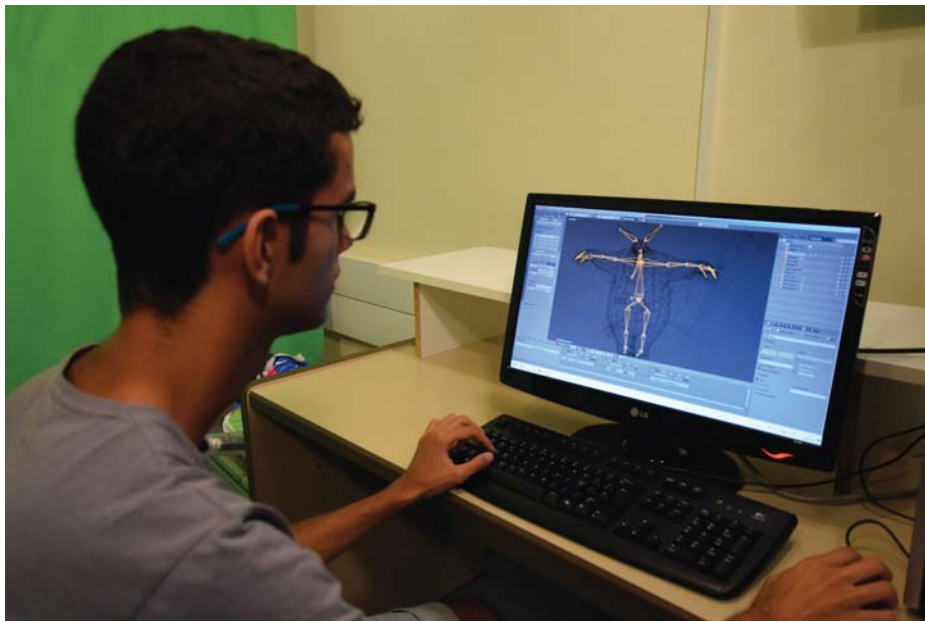


Figura 4.10. Comprobación de un esqueleto.

- **Vídeo demostración**

Para complementar el material de apoyo se ha llevado a cabo la realización de un vídeo en el que se muestra el sistema en funcionamiento y el resultado final.

El proceso para la realización del vídeo ha seguido el esquema de etapas que se muestra en la Figura 4.11.



Figura 4.11. Muestra del proceso seguido para la realización del video demostrativo.

En la primera etapa del proceso, la de preparación, se han conectado y configurado todos los equipos necesarios para la grabación, además de los propios para realizar la captura de movimiento. Los equipos son, los ya nombrados en el apartado de hardware:

- Un dispositivo Kinect de Microsoft para la captura de movimiento.
- Estación de trabajo basada en Intel Pentium Dual Core, 4 GB RAM. Monitor 21” TFT LCD.
- Cámara profesional de vídeo Sony HVR-Z1E.
- Set de estudio con ciclorama e iluminación profesional.

La Kinect se ha colocado a una distancia del individuo de 1,80 metros aproximadamente y a 1 metro del suelo.^[2] Una vez posicionada, se orienta apuntando para la zona donde se situará el individuo. Con la ayuda de la visión que nos proporciona el software Delicode Ni-Mate,^[18] se podrá hacer un buen encuadre de la escena. En las siguientes figuras se muestra como se colocó la Kinect.



Figura 4.12. Orientación de la Kinect.

Una vez bien situada la Kinect en el lugar donde se va a desarrollar la captura, y conectada al ordenador, lo siguiente será calibrarla. Para ello se debe iniciar el software Delicode Ni-Mate y no pasar delante de la cámara hasta pasados unos 5 segundos, después de ejecutar el software. Ya pasado ese tiempo, el individuo se debe poner delante de la Kinect a la distancia marcada, con los brazos en cruz y las piernas a la altura de los hombros. Entonces, en el software aparecerá el mensaje “calibrating...”. Con esto, ya tendremos el sistema listo para hacer la captura. Siempre que el individuo se salga del campo de visión de la Kinect, tendrá que calibrarla antes de empezar con la captura. En la Figura 4.13, se muestra como calibrar el dispositivo.



Figura 4.13. Proceso de calibración de la Kinect.

Se ha utilizado un set de iluminación profesional, compuesto por tres focos Ianiro de luz caliente de 3200 °K, con lámparas de 150 W (Figura 4.14). Para la grabación se ha utilizado únicamente el central, ya que el Kinect no funciona correctamente bajo la luminosidad de los 3 focos a la vez. Por lo tanto, se ha usado sólo uno por establecer un nivel medio de luminosidad entre la captura del sensor Kinect y la videocámara.



Figura 4.14. Set de iluminación.

Para realizar la captura del modelo 3D a la vez que se graba al individuo, se ha utilizado un mezclador de vídeo, el cual grababa la señal de la cámara y la señal de vídeo recibida directamente de la estación de trabajo, mediante un software propio de dicho mezclador, en el disco duro del ordenador.

Una vez realizadas las grabaciones, se comenzó con la fase de edición. Esta fase empezó con un visionado de las capturas realizadas y una selección de las mejores tomas grabadas. Para la edición y montaje del vídeo se utilizó el software Adobe Premiere.^[26] Dicho software de edición de vídeo, nos ofrece las herramientas que se necesitan para crear el video demostrativo.

La idea principal del video, es la de enseñar al lector como debería quedar cada proyecto de Blender a la par que se va avanzando en el tutorial. El video se ha editado de manera que aparezca en un lado la toma grabada del individuo, y en el otro lado, el modelo siguiendo los pasos de aquel. Con esto se muestra cómo el modelo imita los movimientos de la persona real al mismo tiempo, en definitiva, una captura de movimiento en tiempo real.

Para la edición se han utilizado herramientas de cortar, se han creado títulos para cada parte del video, así como un vídeo de presentación del mismo. Además, se genero una barra blanca a modo de línea divisoria entre las dos tomas, para acentuar más que los vídeos son capturados por cámaras diferentes pero al mismo tiempo. A continuación, se muestra una captura del Adobe Premiere mientras se realizaba el proceso de edición.



Figura 4.15. Muestra de momento de edición, donde se observa la división de la pantalla con las dos señales de video unidas.

Por último, se realizó la exportación del vídeo en formato de alta calidad comprimido en MPEG-4 con una resolución de 1280x720p. Este vídeo ha sido almacenado en el CD adjunto a este Trabajo de Fin de Grado.

Capítulo 5.

Pruebas y conclusiones

Para la realización de este Trabajo de Fin de Grado se han realizado distintas pruebas para comprobar el funcionamiento del sistema implementado. En este capítulo se especifican las pruebas realizadas, así como los resultados obtenidos. Además se revisarán los objetivos y se extraerán las diferentes conclusiones y proponiendo posibles mejoras para el futuro.

5.1 Pruebas realizadas

Para comprobar la capacidad del dispositivo Kinect, y del software que lo controla, se han realizado una serie de pruebas para comprobar el comportamiento de ambos. Con estas pruebas se consigue saber cómo se ha de colocar el dispositivo y cuáles son algunas de sus limitaciones. Las pruebas se han realizado, en su totalidad, en el Plató de Televisión de la Escuela de Ingeniería de Telecomunicación y Electrónica. Para el desarrollo de las mismas, se ha contado con una configuración como la que puede observarse en la Figura 5.1. A continuación se pasa a describir dichas pruebas.

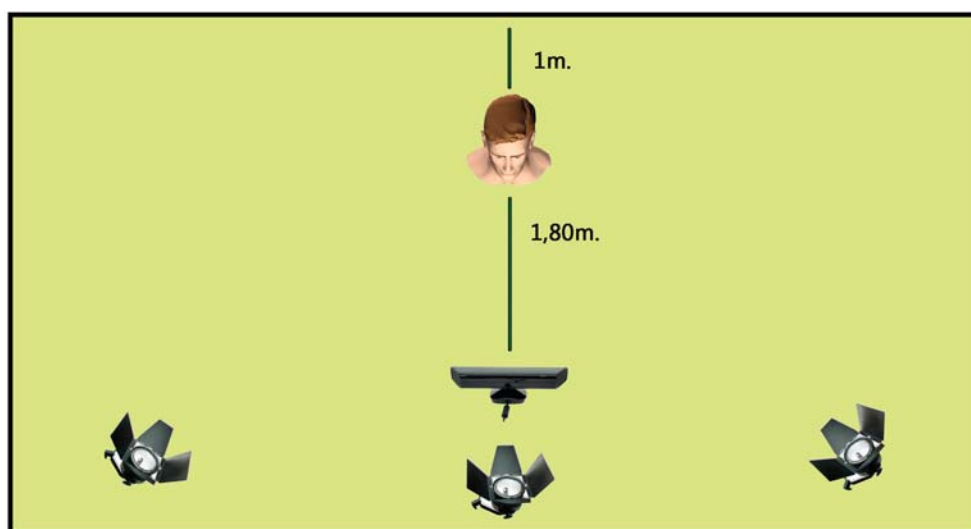


Figura 5.1. Configuración del plató para la realización de la captura de movimiento.

5.1.1 Configuración de la iluminación del entorno

Unos de los parámetros básicos a la hora de obtener una captura de calidad es contar con buenos niveles de iluminación en el recinto, niveles acordes a las especificaciones y limitaciones de los dispositivos de captura. En el caso concreto de Kinect, Microsoft afirma que se puede usar bajo cualquier condición de luz ambiental.^[2]

En esta prueba se ha querido conocer bajo qué condiciones lumínicas trabaja mejor el dispositivo Kinect. Para ello, se han usado los focos del plató y, con la ayuda de la visión del Kinect, que nos proporciona el software Delicode Ni-Mate, se han podido

detectar fallos en la captación del dispositivo cuando el lugar de grabación es muy luminoso, utilizando 3 focos apuntando a la persona (Figura 5.2), menos luminoso, utilizando un único foco de luz apuntando a la persona (Figura 5.3), y sin ningún foco, con luz natural (Figura 5.4).

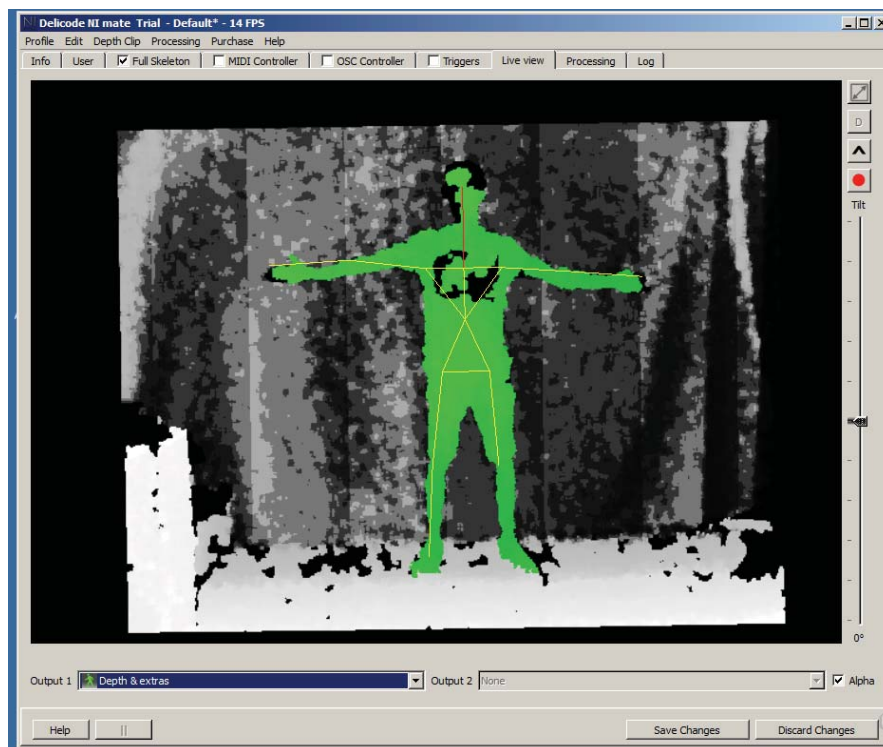


Figura 5.2. Prueba del Kinect bajo la luz de 3 focos.

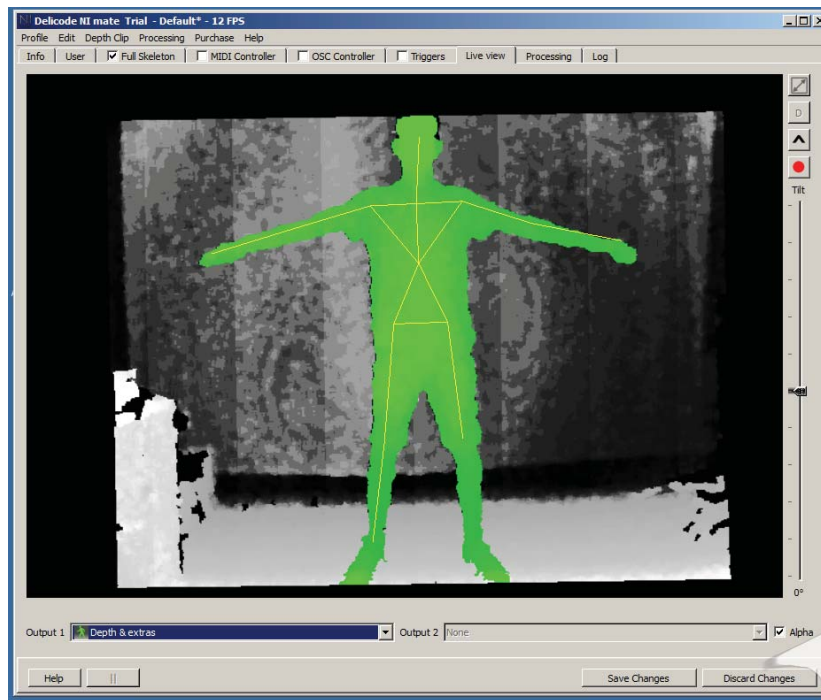


Figura 5.3. Prueba del Kinect bajo la luz de 1 foco de luz.

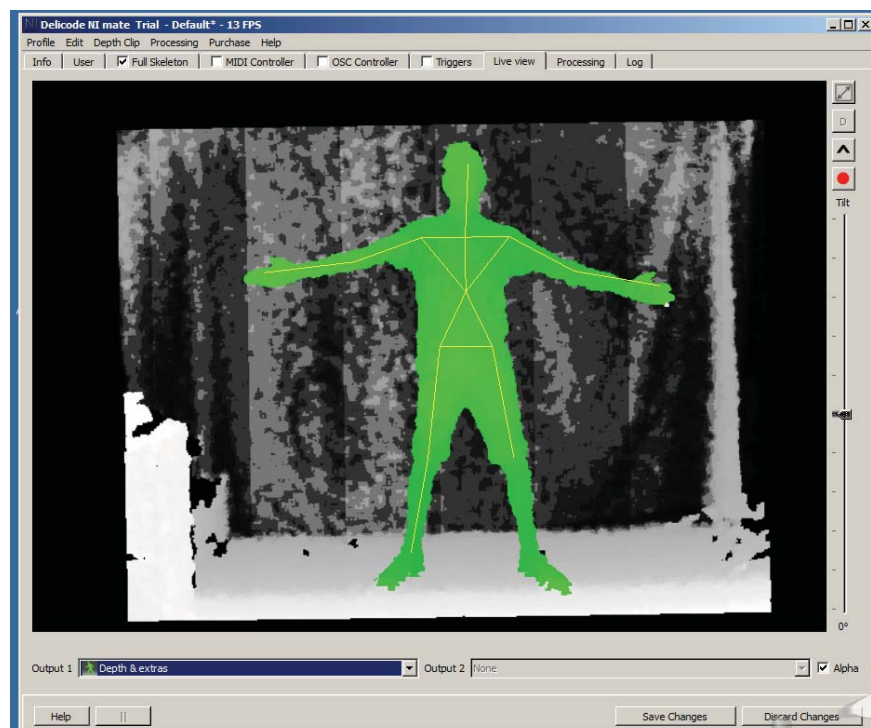


Figura 5.4. Prueba del Kinect con luz natural.

Como se observa en la Figura 5.2, cuanto más luminosidad tenga el lugar donde se desee realizar la captura de movimiento con el dispositivo Kinect, peor será la captación de este. Se puede apreciar que en la zona del pecho aparecen unas manchas negras iguales que

las que la Kinect detecta como fondo. Además, si se presta atención al esqueleto que automáticamente se coloca siguiendo las articulaciones de la persona que se coloca delante, se puede observar que la línea que el dispositivo referencia por la zona de la cabeza, aparece de color rojo. Esto significa que no detecta la cabeza.

En el caso de la Figura 5.3, se ha realizado la prueba emplean un solo foco y se obtienen mejores resultados. Ya no aparecen las manchas negras, pero en el esqueleto guía que genera se aprecia una discontinuidad en el dibujo de las líneas, por lo que el Kinect detecta a la persona pero esto podría suponer que el personaje realizará algún movimiento que la persona real no realice. En cambio, en la Figura 5.4, no aparecen manchas negras y las líneas del esqueleto guía se aprecian más continuas.

5.1.2 Campo y entorno visual

En esta prueba se ha analizado que sucede cuando nos salimos del campo de visión del dispositivo Kinect. Además, se ha comprobado a cuanta distancia de la pared es la idónea estar para que la captura sea buena y que falle lo menos posible (se considera fallo al momento en el que la cámara no detecta bien una articulación o el esqueleto guía se sale de la forma del cuerpo de la persona real).

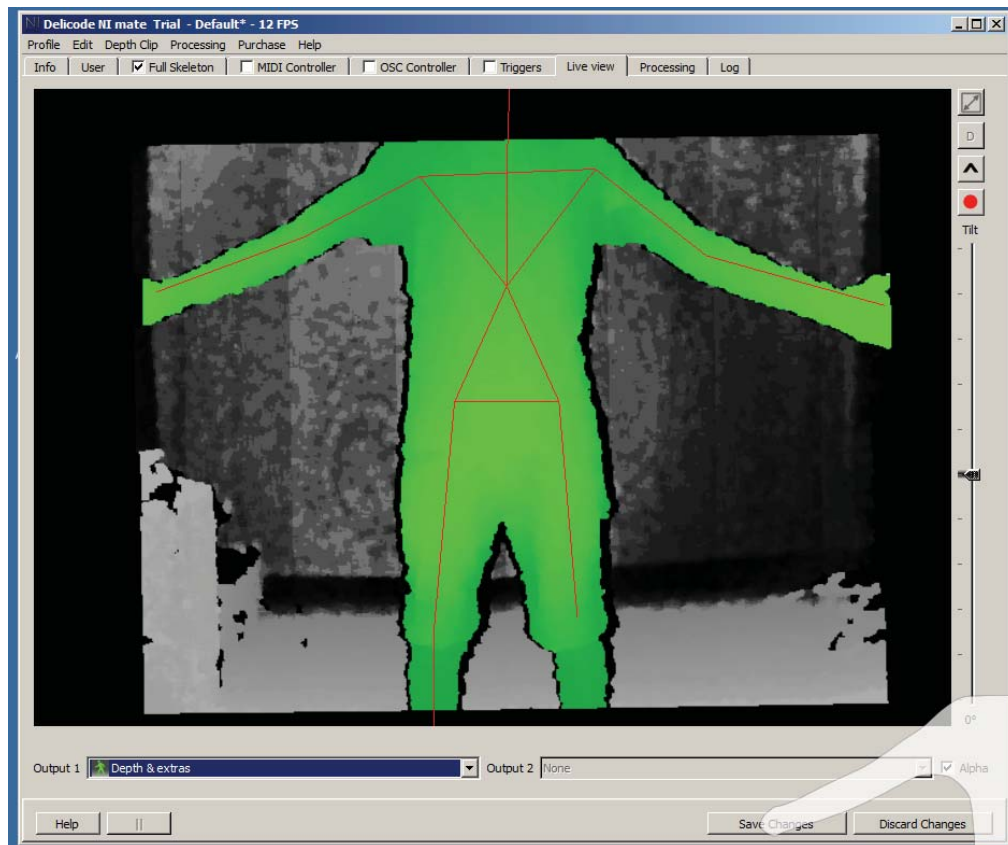


Figura 5.5. Prueba en la que el individuo se acerca demasiado al dispositivo Kinect.

En la Figura 5.5, se observa que, al acercarnos demasiado, el esqueleto guía se vuelve rojo totalmente, ya que ha perdido en su campo de visión los cuatro extremos, es decir, manos, pies y la cabeza.

A continuación, en la Figura 5.6, se ha hecho la misma prueba pero cambiando el fondo de la pared de detrás por uno blanco. Como la camiseta, que la persona llevaba, era blanca, se quería comprobar si es importante que haya un contraste entre la pared del fondo y la ropa del individuo. En la prueba anterior, con el fondo verde, el esqueleto guía se volvía rojo pero no se salía del contorno del cuerpo de la persona que se está grabando, mientras que con el fondo blanco, al tener la camiseta blanca, el esqueleto guía se sale del cuerpo del individuo.

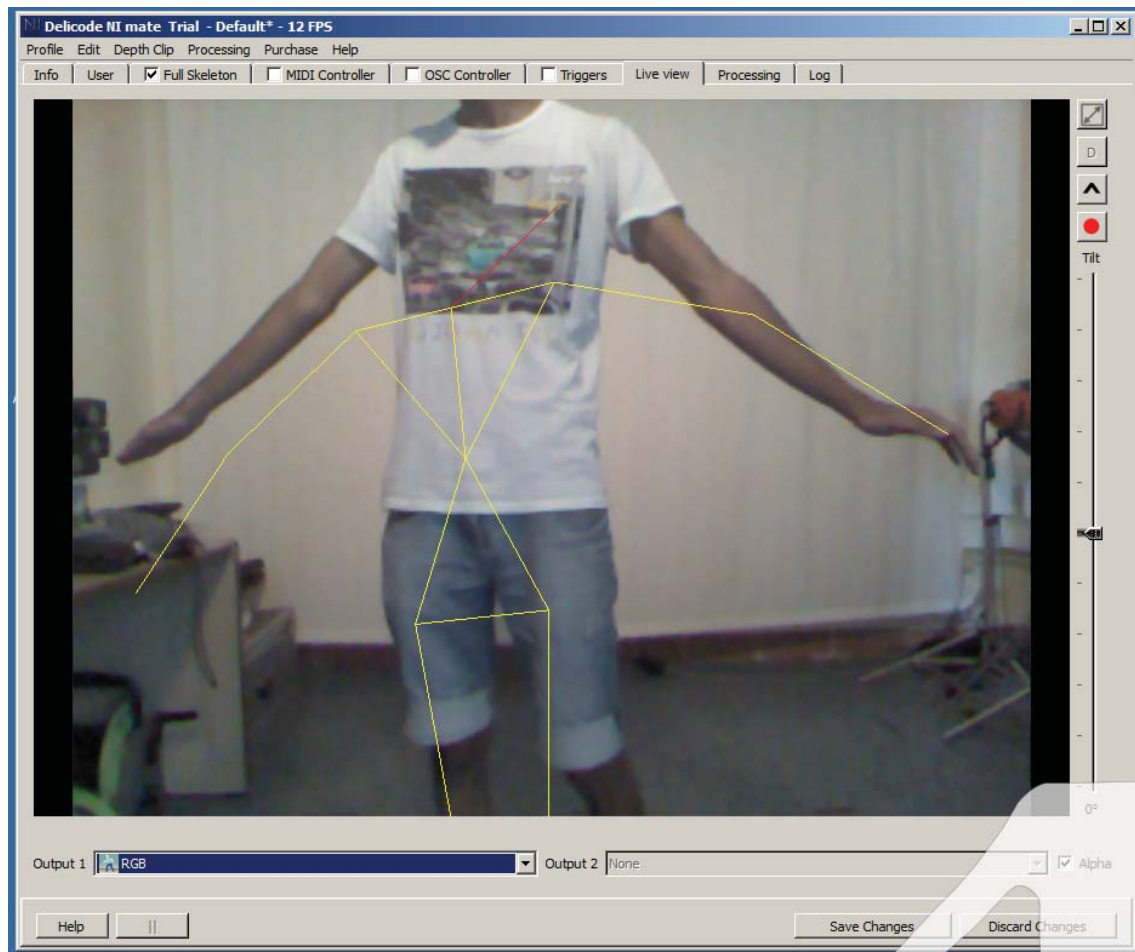


Figura 5.6. Prueba en la que el individuo se acerca demasiado al dispositivo Kinect, pero esta vez con el fondo de color blanco. Analizando si es importante que haya un contraste entre el fondo y la ropa del individuo.

Además se ha querido observar cómo se comporta el sensor Kinect cuando el individuo se pega a la pared de detrás. Se ha realizado una prueba con fondo blanco y otra con fondo verde. Las dos pruebas han dado muy malos resultados de captura con lo que se comprueba que hay que estar lo más separado posible de la pared, aproximadamente 1 metro que es lo que especifica el fabricante. A continuación, se muestran en la Figura 5.7 con fondo blanco y, en la Figura 5.8, con fondo verde.

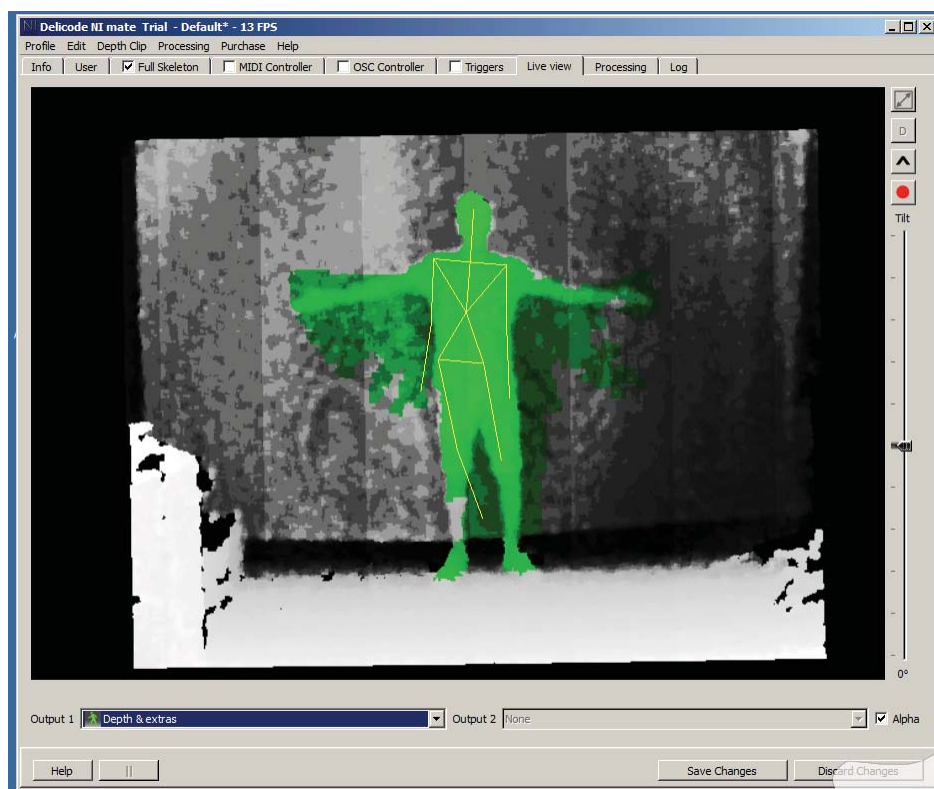


Figura 5.7. Prueba del individuo en contacto con la pared de fondo blanco.

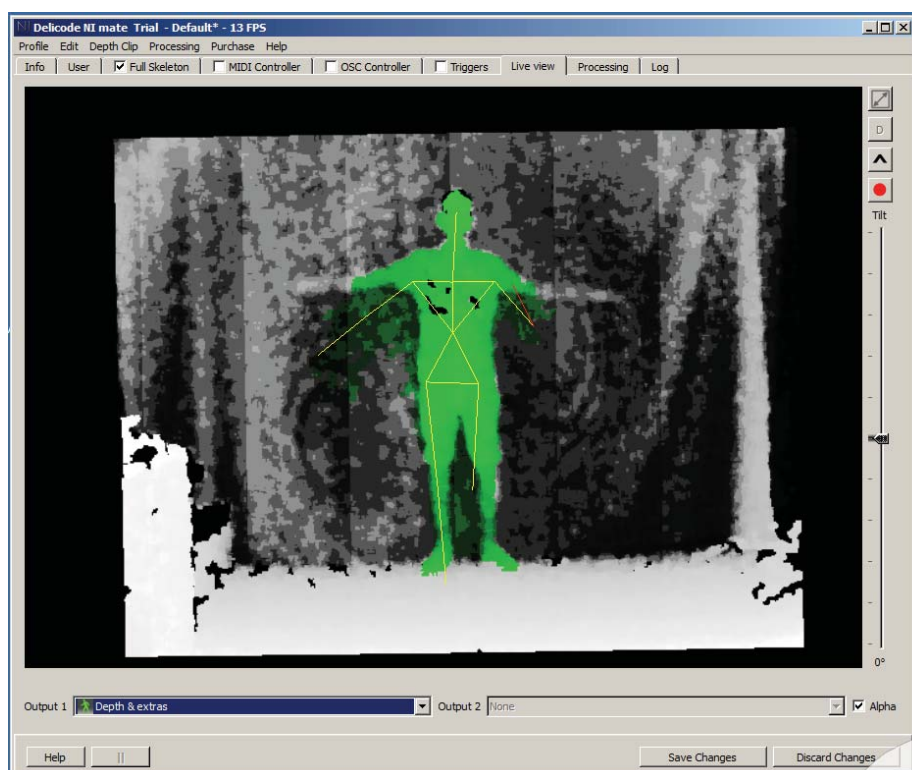


Figura 5.8. Prueba del individuo en contacto a la pared de fondo verde.

5.2 Conclusiones

Gracias a la realización de este Trabajo de Fin de Grado se ha logrado profundizar un poco más en el mundo de la animación 3D. Además, se ha encontrado la manera de realizar captura de movimiento sin tener que disponer de una tecnología muy cara. Como se ha podido comprobar en este trabajo, el dispositivo Kinect es un dispositivo que no sólo es válido para juegos.

Se han cumplido los dos objetivos al 100%, además de aprender a utilizar el software Blender y el software Delicode Ni-Mate para el control del dispositivo Kinect.

Para el futuro:

- Se podrían añadir más dispositivos Kinect e implementarlos para tener una ampliación en la percepción.
- Implementar el reconocimiento, no sólo de las articulaciones principales, sino también el de los dedos de las manos y los pies.
- Implementar el reconocimiento de expresiones faciales.

BIBLIOGRAFÍA

En este apartado se expone la bibliografía consultada para la realización de este Trabajo de Fin de Grado

- [1] Windsor, Brian, and Kitagawa, Midori. *MoCap for Artists Workflow and Techniques for Motion Capture*, 2008, Focal Press.
- [2] Manual de Kinect. <http://www.xbox.com/es-ES/Kinect> (Fecha de última visita: 28/01/2014)
- [3] Brekelmans, Jasper. Proyecto Brekel Kinect (2010). Sitio web oficial: <http://www.brekel.com> (Fecha última visita: 17/03/2014)
- [4] López Contreras, O. (2012). Motion Capture con Microsoft Kinect (Proyecto Fin de Carrera). Universidad Autónoma de Barcelona, España. Recuperado de: http://www.recercat.net/bitstream/handle/2072/212584/PFC_OriolLopezContreras.pdf?sequence=1
- [5] Ejemplo de proyecto de diagnóstico de desórdenes infantiles basado en kinect. <http://revistaradiologica.blogspot.com.es/2012/06/el-uso-potencial-de-kinect-en-salud-y.html> (Fecha última visita: 15/02/2014)
- [6] Ejemplo de proyecto de rehabilitación médica basado en kinect. <http://www.virtualrehab.info/es/> (Fecha última visita: 15/02/2014)
- [7] Ejemplo de proyecto de teleasistencia médica basado en kinect. http://www.ilitia.com/Expertise/Teleasistencia_basada_en_Kinect_Teki (Fecha última visita: 15/02/2014)
- [8] Sitio web oficial de Blender. <http://www.blender.org> (Fecha última visita: 15/02/2014)
- [9] Hess, Roland. *Blender*, 2010, Focal Press.
- [10] Ejemplo de película en la que se usó Blender. <http://mango.blender.org> (Fecha de última visita: 15/02/2014)
- [11] Ejemplo de película de animación hecha con Blender. <http://www.sintel.org> (Fecha de última visita: 16/02/2014)
- [12] Ejemplo de videojuego hecho con Blender. <http://www.blendernation.com/2012/04/25/game-z-virus-2/> (Fecha de última visita: 16/02/2014)
- [13] Información acerca del descubrimiento de Eadweard Muybridge. http://www.hrc.utexas.edu/exhibitions/permanent/windows/southeast/eadweard_muybridge.html (Fecha de última visita: 20/05/2014)
- [14] Información acerca de la técnica de la rotoscopia. <http://animacionartesanatecnicas.blogspot.com.es/2010/04/roscopia.html> (Fecha de última visita: 20/05/2014)
- [15] Barbara Robertson, "Mike, the talking head", 1988, Computer Graphics World.

- [16] Jeff Kleiser, "Character motion systems," Course Notes: Character Motion Systems, ACM SIGGRAPH 93, Anaheim, CA, 1993.
- [17] Tecnologías de captura de movimiento.
<http://patentscope.wipo.int/search/en/detail.jsf?jsessionid=17E763B21A1E3D3DC29428ED0B00B660.wapp2nB?docId=WO2003079672&recNum=1&maxRec=&office=&prevFilter=&sortOption=&queryString=&tab=PCTDescription> (Fecha de última visita: 02/06/2014)
- [18] Sitio web del software Delicode Ni-Mate. <http://www.ni-mate.com> (Fecha de última visita: 02/06/2014)
- [19] Sitio web del software Autodesk 3Ds Max. <http://www.autodesk.es/products/3ds-max/overview> (Fecha de última visita: 05/06/2014)
- [20] Sitio web del software Autodesk Maya. <http://www.autodesk.es/products/autodesk-maya/overview> (Fecha de última visita: 05/06/2014)
- [21] Sitio web del software Cinema 4D. <http://www.maxon.net/es/products/cinema-4d-studio.html> (Fecha de última visita: 05/06/2014)
- [22] Sitio web del software Google Sketchup. <http://www.sketchup.com/es/products/sketchup-pro> (Fecha de última visita: 05/06/2014)
- [23] Sitio web del software Wings 3D. <http://www.wings3d.com> (Fecha de última visita: 05/06/2014)
- [24] Sitio web del software K-3D. <http://www.k-3d.org> (Fecha de última visita: 05/06/2014)
- [25] Sitio web del software Adobe Photoshop. <http://www.adobe.com/es/products/photoshop.html> (Fecha de última visita: 12/06/2014)
- [26] Sitio web del software Adobe Premiere. <http://www.adobe.com/es/products/premiere.html> (Fecha de última visita: 12/06/2014)
- [27] Sitio web del software Fraps. <http://www.fraps.com> (Fecha de última visita: 14/06/2014)

P1. PRESUPUESTO

Don Néstor Manuel Alemán Soler, autor del presente Trabajo de Fin de Grado, declara que:

El Trabajo de Fin de Grado con título “Solución de bajo coste de captura de movimiento basada en Kinect”, desarrollado en la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de las Palmas de Gran Canaria, tiene un coste de desarrollo total de **16.403,62** correspondiente a la suma de las cantidades consignadas a los apartados considerados a continuación.

Las Palmas de Gran Canaria a 18 de julio de 2014

Firma: Néstor Manuel Alemán Soler

1. Presupuesto detallado

El presupuesto de este Trabajo de Fin de Grado que se va a detallar a continuación es orientativo y debido a que en la página web del COITT han suprimido las listas de los Honorarios Orientativos, se va a calcular en función de las listas publicadas en el COIT.

Hasta el año 2008, el COIT ha venido publicando unas listas de Honorarios Orientativos que en los dos últimos años se denominaron Costes Estimados de Trabajos Profesionales. Por modificación de la Ley de Colegios Profesionales, mediante Ley 25/2009 de 22 de diciembre, no es posible seguir publicando estas listas. El Colegio no puede elaborar baremos de honorarios, ni siquiera orientativos, salvo que sean con la finalidad de tasar costes en los procedimientos judiciales. No obstante y con absoluto respeto a lo establecido en la mencionada Ley, el COIT dispone de una herramienta de cálculo para que el colegiado pueda valorar por sí mismo los trabajos profesionales que realiza, en función de la organización de su actividad. Esta herramienta se encuentra en la web del COIT en Ejercicio Profesional / Apoyo y Desarrollo Técnico / Información General.

Siguiendo éstas recomendaciones del Colegio oficial de Ingenieros de Telecomunicación (COIT) que nos ofrece en esta herramienta orientativa para trabajos profesionales en 2011, el presupuesto se ha desglosado en varias secciones en las que se han separado los distintos **costes orientativos** asociados al desarrollo del proyecto. Estos costes se dividen en:

1. Recursos materiales.
2. Trabajo tarifado por tiempo empleado.
3. Costes de redacción del Trabajo Fin de Grado.
4. Material fungible.
5. Derechos de visado del COIT.
6. Costes de tramitación y envío.
7. Aplicación de impuestos.

2. Recursos materiales

Para la ejecución de este Trabajo Fin de Grado han sido necesarios las herramientas software para la realización de la captura de movimiento y el control del dispositivo de captura, los softwares para la captura y el retoque fotográfico y la edición de video, así como el paquete office para la redacción de la memoria. En cuanto al hardware que se ha necesitado, se encuentran el dispositivo Kinect, una videocámara profesional, un set de estudio con ciclorama e iluminación profesional, un ordenador y una impresora.

La amortización se calcula sobre el tiempo útil del recurso material. El sistema de amortización se toma como lineal, asumiendo que el inmovilizado material se desprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula usando la siguiente fórmula:

$$\text{Cuota} = \frac{\text{Valor de adquisición} - \text{Valor residual}}{\text{Tiempo de vida útil}}$$

a. Recursos Hardware

Para la ejecución de este estudio las herramientas hardware que se han utilizado son las siguientes:

- Un dispositivo Kinect de Microsoft.
- Estación de trabajo basada en Intel Pentium Dual Core, 4 GB RAM. Monitor 21” TFT LCD.
- Cámara profesional de vídeo Sony HVR-Z1E.
- Trípode MANFROTTO Manfrotto-Estabilizador Fig Rig+remoto
- Set de estudio con ciclorama e iluminación profesional.
- Mezclador Tricaster 450 Extreme.
- Ordenador personal.
- Impresora.

Recurso	Valor de adquisición (€)	Valor residual (€)	Vida útil (años)	Cuota anual (€)	Uso (meses)	Cuota aplicable (€)
Dispositivo Kinect	119	50	5	13,8	4	4,6
Estación de trabajo	936,85	400	5	107,37	4	35,79
Cámara profesional	5.142,50	2000	8	392,81	4	130,93
Trípode MANFROTTO	676,10	400	4	69,02	4	23
Set de estudio de iluminación	1200	500	10	70	4	23,33
Mezclador Tricaster	19.795,50	10.000	12	816,29	4	272,09
Ordenador personal	1039	400	4	159,75	4	53,25
Impresora	90	30	3	20	4	6,66
TOTAL						549,65€

Tabla P1. Recursos Hardware.

b. Recursos Software

Las herramientas Software necesarias son:

- Software de modelado y animación 3D Blender.
- Software para el control del sensor y realización de la captura Delicode Ni-Mate.
- Adobe Photoshop CS6.
- Adobe Premiere CS6.
- Fraps.
- Microsoft Office 2013.

Recurso	Valor de adquisición (€)	Valor residual (€)	Vida útil (años)	Cuota anual (€)	Uso (meses)	Cuota aplicable (€)
Blender	0	0	5	0	4	0
Delicode Ni-Mate	0	0	4	0	4	0
Adobe Photoshop CS6	942,18	300	5	128,43	4	42,81
Adobe Premiere CS6	1.060,26	400	5	132,052	4	44,01
Fraps	27	5	2	11	4	3,66
Microsoft Office 2013	152,99	75	3	25,99	4	8,66
TOTAL						99,14€

Tabla P2. Recursos Software.

3. Trabajo tarifado por tiempo empleado.

En este Trabajo Fin de Grado se ha invertido 300 horas en las tareas de formación, especificación, desarrollo y documentación necesarias para la elaboración del mismo. El importe de las horas de trabajo empleadas para la realización del proyecto se calcula siguiendo las recomendaciones del COIT:

$$H = C_t * 74,88 * H_n + C_t * 96,72 * H_e$$

Donde:

- H son los honorarios totales por el tiempo dedicado.
- H_n son las horas normales trabajadas (dentro de la jornada laboral).
- H_e son las horas especiales.
- C_t es un factor de corrección función del número de horas trabajadas.

Para la realización de este Trabajo Fin de Grado se han necesitado 300 horas (3h/día * 100 días), todas ellas dentro del horario normal.

Según el COIT, el coeficiente C_t tiene un valor variable en función del número de horas empleadas de acuerdo con la siguiente tabla:

Horas empleadas	Factor de corrección C_t
Hasta 36 horas	1,00
Desde 36 a 72 horas	0,90
Desde 72 a 108 horas	0,80
Desde 108 a 144 horas	0,70
Desde 144 a 180 horas	0,65
Desde 180 a 360 horas	0,60
Desde 360 a 540 horas	0,55

Tabla P3. Factor de corrección según el COIT.

Como se puede observar el número de horas está comprendido en más de 180 horas, por lo que según la Tabla P1.4 el factor de corrección es de 0.60. Con ello, la ecuación del importe de horas de trabajo resulta de la siguiente forma:

$$H = 0.60 * 74,88 * 300 + 0.60 * 96,72 * 0 = 13.478,40\text{€}$$

Los honorarios totales por tiempo dedicado libres de impuestos ascienden a: *trece mil cuatrocientos setenta y ocho euros con 40 céntimos (13.478,40 €)*.

4. Costes de redacción del Trabajo Fin de Grado

El importe de la redacción del proyecto se calcula de acuerdo a la siguiente expresión:

$$R = 0.07 * P * C_n$$

Donde:

- P es el presupuesto del proyecto
- C_n es el coeficiente de ponderación en función del presupuesto.

En la siguiente tabla se muestra el presupuesto calculado hasta el momento:

Recursos	Costes
Recursos Software	99,14€
Recursos Hardware	549,65€
Trabajo tarifado por tiempo empleado	13.478,40€
TOTAL	14.127,19€

Tabla P4. Presupuesto total hasta el momento

El presupuesto calculado hasta el momento asciende a 14.127,19 €. Como el coeficiente de ponderación para presupuestos menores de 30.050 € viene definido por COIT con un valor de 1,00 el coste derivado de la redacción del trabajo es de:

$$R = 0.07 * 14.127,19 * 1 = 988,90€$$

El importe por redacción de trabajo asciende a la cantidad de: *novecientos ochenta y ocho euros con noventa céntimos (988,90€)*

5. Material fungible

Materiales	Costes
Folios	12€
Tóner de impresora	100€
Encuadernación	5€
TOTAL	117€

Tabla P5. Material Fungible.

6. Derechos de visado del COIT

Los gastos de visado del COIT, se tarifican mediante la siguiente ecuación:

$$V = 0.006 * P * C_V$$

Donde:

- P es el presupuesto del proyecto.
- C_V es el coeficiente reductor en función del presupuesto del proyecto.

El presupuesto de ejecución material, calculado hasta el momento asciende a la cantidad de: 15.233,09 €

El coeficiente C_V dado por el COIT para presupuestos menores de 30.050 € tiene el valor de 1,00 por lo que:

$$V = 0.006 * 15.233,09 * C_V = 91,39€$$

El coste de los derechos de visado del trabajo asciende a la cantidad de: *noventa y un euros con treinta y nueve céntimos (91,39€)*.

7. Gastos de tramitación y envío

Los gastos de tramitación y envío están fijados en 6,01 €.

8. Aplicación de impuestos

Para la actividad económica del presente trabajo el valor del Impuesto General Indirecto Canario (I.G.I.C.) graba el presupuesto con un 7%. El coste total del trabajo con el I.G.I.C. incluido se desglosa en la siguiente tabla:

Descripción	Subtotal (€)
Recursos materiales	
Hardware	549,65
Software	99,14
Trabajo tarificado por tiempo empleado	13.478,40
Costes de redacción del trabajo	988,90
Material fungible	117
Derechos de visado del COIT	91,39
Gastos de tramitación y envío	6,01
Suma (€)	15.330,49
IGIC 7% (€)	1.073,13
TOTAL (€)	16.403,62

Tabla P6. Coste total del trabajo.

El importe final al que asciende el presupuesto de este trabajo es de: *dieciséis mil cuatrocientos tres euros con sesenta y dos céntimos (16.403,62€)*.

Las Palmas de Gran Canaria a 18 de julio de 2014

Firma: Néstor Manuel Alemán Soler

ANEXO I

Tutorial sobre cómo realizar captura de movimiento a través de datos externos y en tiempo real

Este tutorial consta de 3 partes, en las que, al finalizarlo, adquirirá unos conocimientos básicos sobre el MOCAP, y los softwares utilizados para su realización.

Tutorial Parte 1.

Instalación de los softwares y familiarización con la interfaz de Blender

En esta parte del tutorial se muestra el proceso de descarga e instalación de los softwares a utilizar, así como, una pequeña guía para la familiarización con el software Blender.

1. Descarga del software necesario para la realización de MOCAP

Lo primero que se debe hacer es dirigirse al sitio web del software de modelado 3D y animación Blender, “www.blender.org” como se muestra en la Figura T1.1.

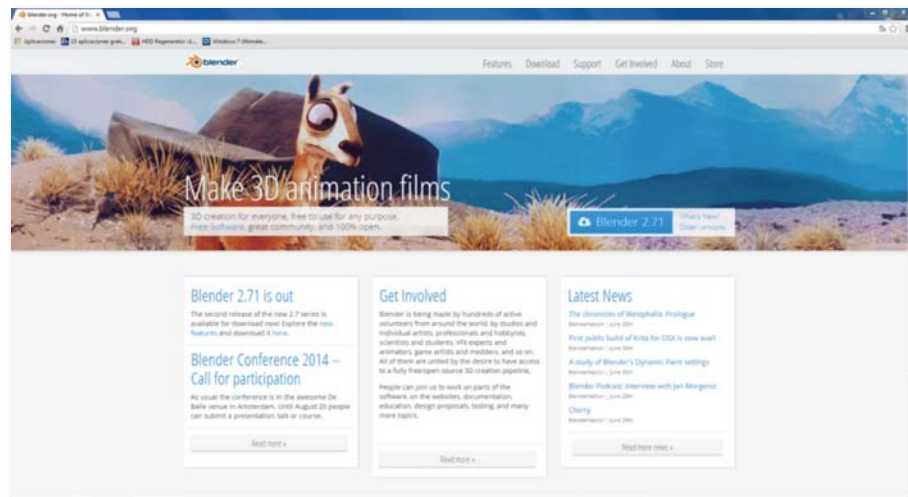


Figura T1.1. Página principal del sitio web de Blender

Mediante el enlace “Download”, arriba a la derecha, se iniciará el proceso de descarga del fichero de instalación.

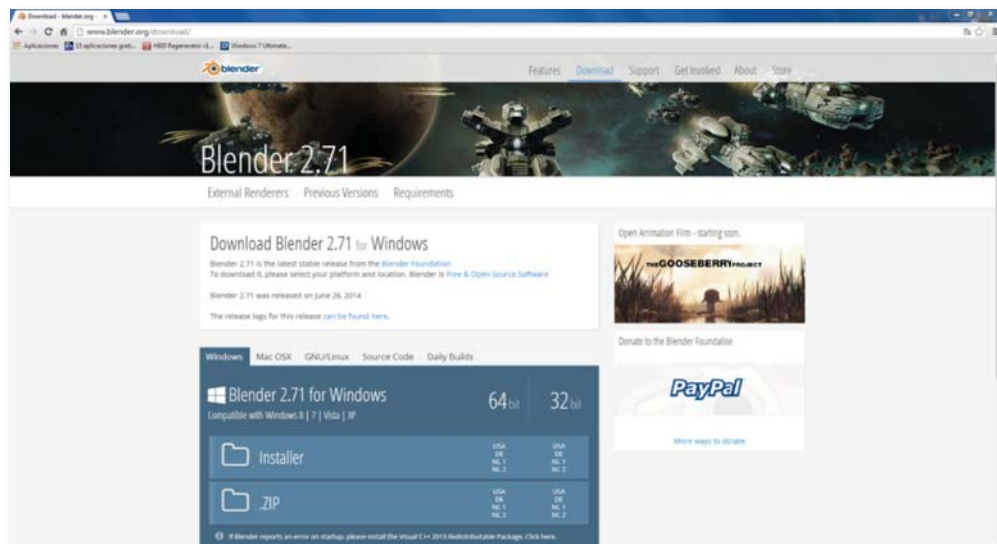


Figura T1.2. Lugar de descarga de Blender

Una vez en ella, al bajar, aparecerá una tabla, como la que se muestra en la Figura T1.3, donde se podrá descargar el software. Se deberá pulsar en el enlace adecuado a su sistema operativo.

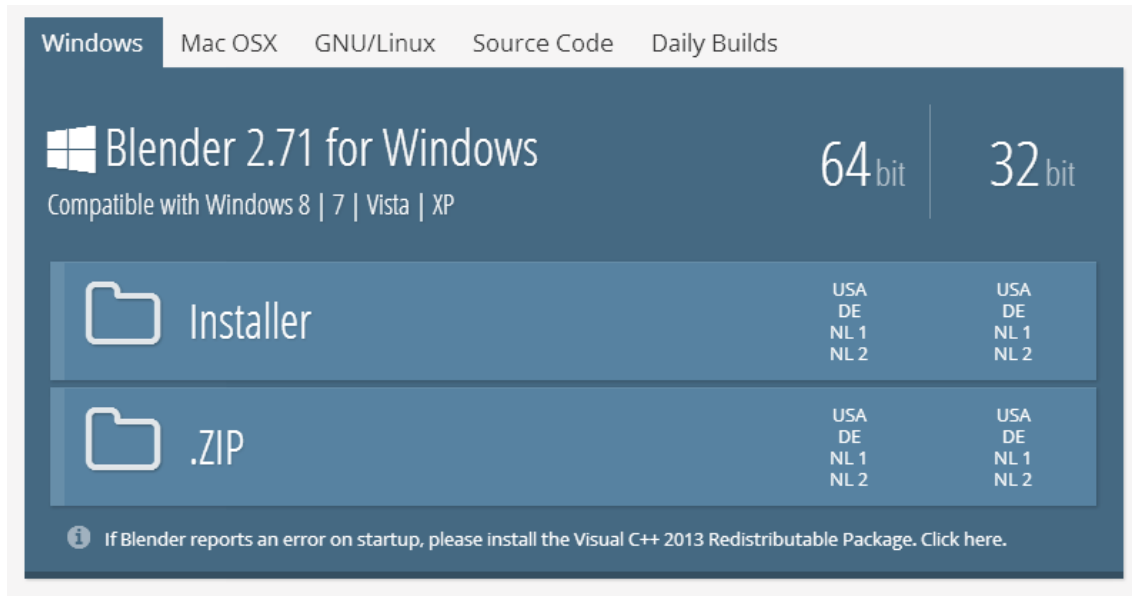


Figura T1.3. Tabla de descarga de la versión según sistema operativo

Se podrá descargar el archivo de instalación e instalarlo de raíz en su ordenador, o una carpeta “.zip”, la cual se podrá guardar en una memoria FLASH y abrirlo en cualquier ordenador sin necesidad de instalarlo.

Lo siguiente que se hará, será descargar el software que controlará la Kinect. Al igual que antes, se dirigirá a la página del software, “ www.ni-mate.com”, y aparecerá la página principal tal y como se muestra en la Figura T1.4.

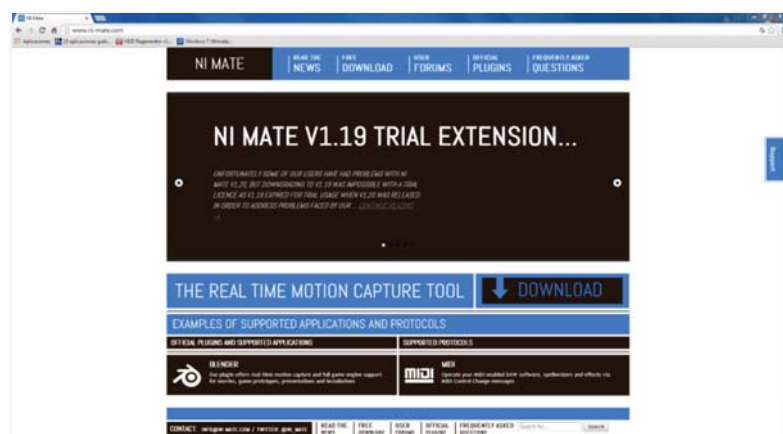


Figura T1.4. Página principal del sitio web de NI-MATE

En la que, tal y como se hizo antes, se pulsará “FREE DOWNLOAD”, situado, arriba a la derecha, que le dirigirá a la siguiente página (Figura T1.5).



Figura T1.5. Lugar de descarga de NI-MATE

En ella, al bajar, se encontrará con una tabla, como la que se muestra en la Figura T1.6, donde se podrá descargar el software. Se pulsará en el enlace adecuado a su sistema operativo.

SYSTEM	DOWNLOAD
WINDOWS	
Windows XP, Vista, 7, 8 32/64 bit	↓ DOWNLOAD
Windows XP, Vista, 7, 8 32/64 bit (with Kinect for Windows support)	↓ DOWNLOAD
MAC OS X (INTEL)	
Mac OS X 10.6+	↓ DOWNLOAD
Mac OS X 10.6+ (OpenNI 2 version)	↓ DOWNLOAD
GNU/LINUX	
Ubuntu 12.04 64 bit	↓ DOWNLOAD
Ubuntu 12.04 32 bit	↓ DOWNLOAD

Figura T1.6. Tabla de descarga de la versión de NI-MATE según sistema operativo

Una vez descargado e instalado habrá que pedir una licencia para poder usarlo. Existen dos posibilidades, o una licencia completa o, la que se escogerá en este caso, una licencia de prueba, la cual no tiene límite de uso. El único inconveniente es que cada cierto tiempo saldrá por pantalla el logo de la empresa que lo desarrolla. Se tendrán que rellenar los datos que se requieren para obtener la licencia, la cual se recibirá por correo electrónico. Para validarla se tendrá que almacenar en una carpeta del ordenador la cual se le indicará al software al iniciarlo.

2. Familiarización con el entorno Blender

Una vez instalado el Blender, se hará un recorrido por su interfaz para la familiarización con el entorno. Lo primero que se encontrará, al ejecutarlo, es lo que se muestra en la siguiente figura (Figura T1.7):

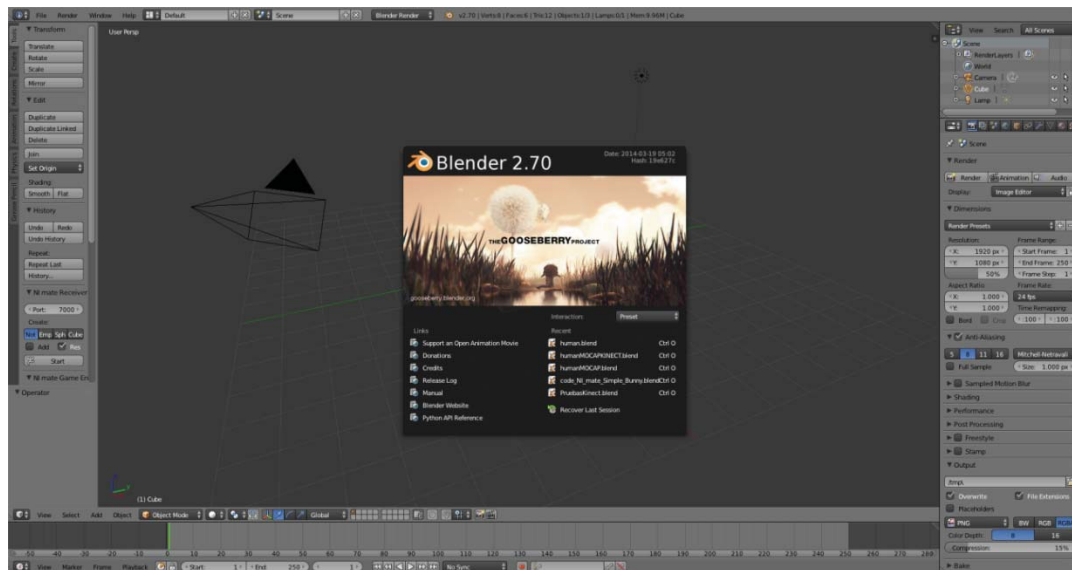


Figura T1.7. Ventana principal al iniciar Blender

En la ventana central (Figura T1.8) tendrá, arriba a la izquierda el número de la versión de nuestro software. En la columna de la izquierda, muestra una serie de enlaces a la ayuda, tutoriales para su uso o si se quiere hacer alguna donación, ya que se trata de software libre. En la columna de la derecha abajo, los proyectos que se han abierto recientemente. Por último, se dispone de un desplegable por si viene de usar otros programas de modelado 3D, 3DS MAX y MAYA, para que se le aplique la forma de interaccionar predeterminada con el software Blender de la misma manera que uno de los mencionados.

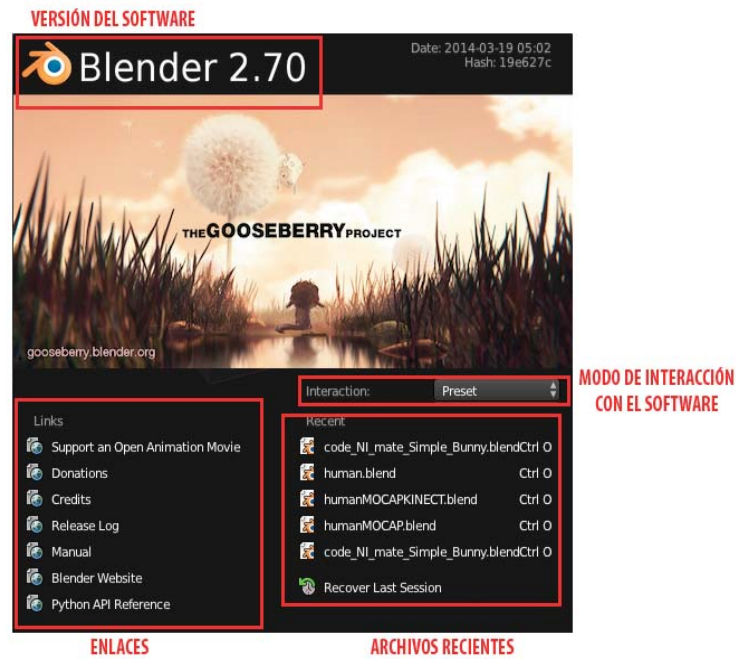


Figura T1.8. Ventana central al iniciar Blender

Para quitar esta pantalla, se pinchará fuera de ella y aparecerá, por defecto, la siguiente pantalla (Figura T1.9):

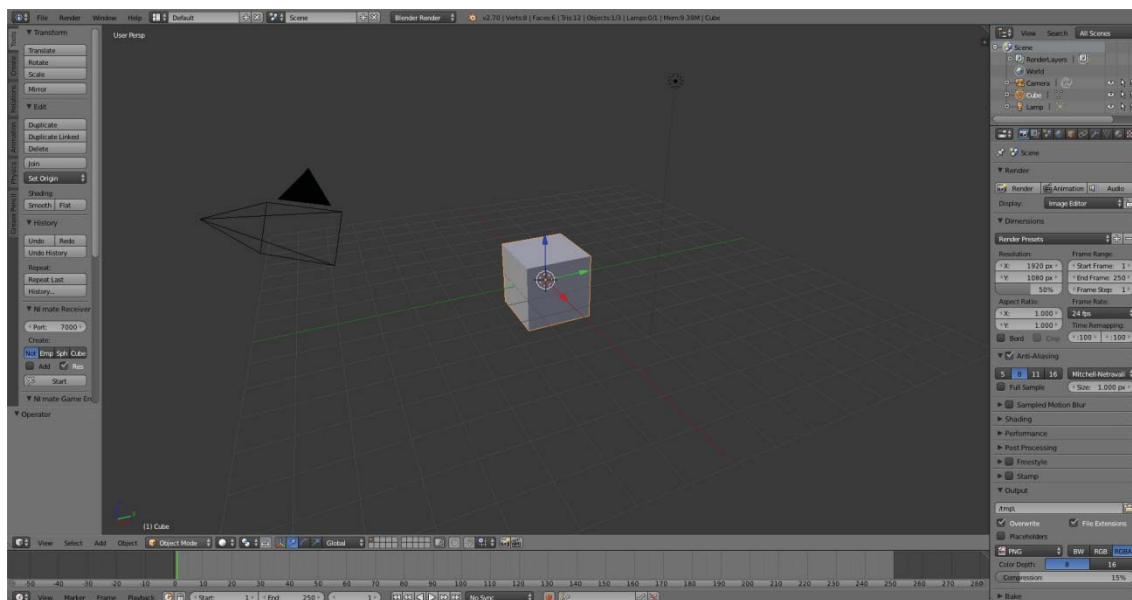


Figura T1.9. Pantalla principal por defecto de Blender

Esta será nuestra pantalla principal, por defecto, siempre que se abra un proyecto nuevo. La interfaz de Blender se basa en ventanas no solapadas. Programas como Photoshop o After Effects, están basados en paletas flotantes, las cuales se deben mover al

gusto de cada uno para que molesten lo menos posible a la hora de trabajar en un proyecto. Blender evita este problema prohibiendo que se solapen. La zona de visualización de Blender se puede dividir en todas las secciones que deseemos pero ninguna se solapa.

2.1 Modificar el tamaño de las ventanas o vistas

En Blender, los bordes entre las distintas ventanas o vistas se pueden mover desplazando el ratón sobre dicho borde. Al hacerlo, el cursor se convierte en una flecha doble. Al pulsar y arrastrar el borde, se cambiará el tamaño de la ventana. En la Figura T1.10 se puede observar un ejemplo de ello.

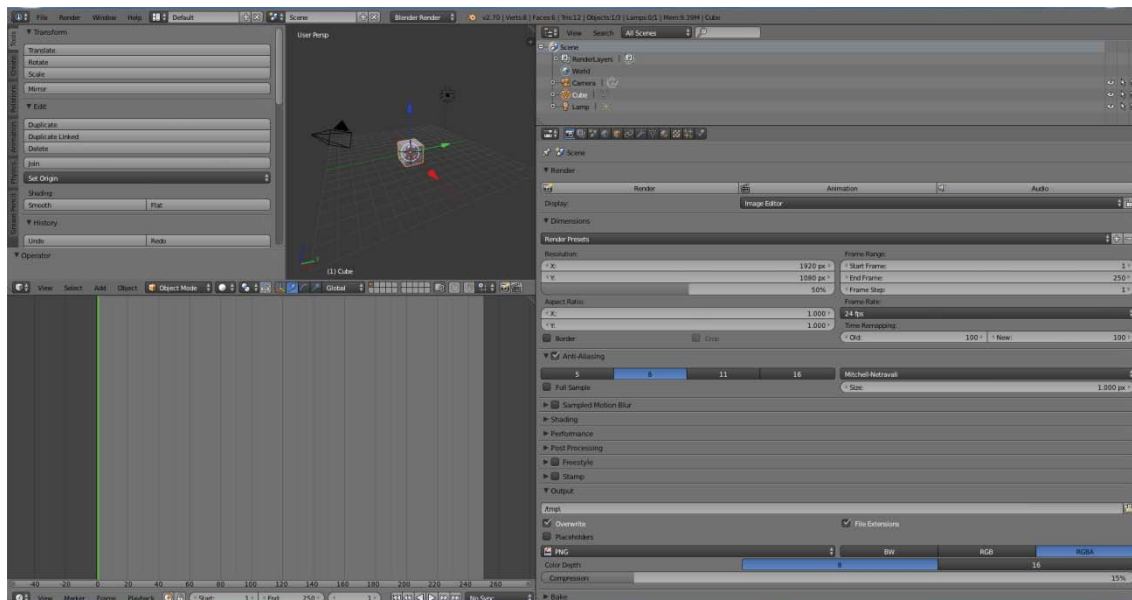


Figura T1.10. Ejemplo de modificación del tamaño de las vistas en Blender

2.2 Los encabezados

En las ventanas se podrán encontrar encabezados. Un encabezado contiene controles específicos de los contenidos de la ventana. En la Figura T1.11 se muestra un encabezado. En los encabezados se selecciona lo que se precisa tener en una ventana, y, con ello, se mostrarán las herramientas de uso frecuente para esa opción. Las ventanas pueden tener o no encabezado, ya que, para maximizar el espacio útil de pantalla, se tiene la posibilidad de ocultarlos desplazando el cursor sobre el límite entre el encabezado y el resto de la pantalla. Al hacerlo, aparecerá un pequeño símbolo +, que, al pulsarlo, volverá a aparecer el encabezado.



Figura T1.11. Ejemplo de encabezado en Blender

En el extremo izquierdo de cada encabezado, aparece un icono que mostrará el tipo de ventana. Si se pincha en ese icono, se desplegará un menú con las opciones posibles (Figura T1.12). Con esto, cualquiera de las ventanas de la pantalla actual, se podrá configurar para mostrar cualquiera de estas opciones en cualquier momento.

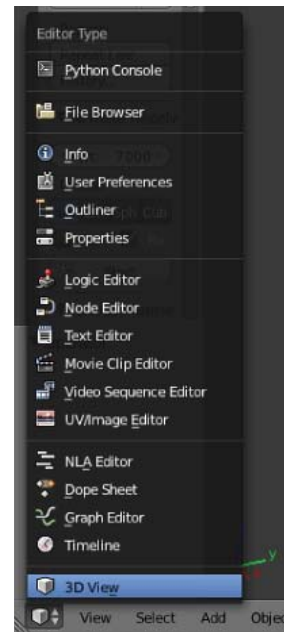


Figura T1.12. Menú desplegable con las opciones de ventana posibles

2.3 Zona de acción de ventanas

Por debajo y a la izquierda del icono de tipo de ventana, se aprecian unas líneas diagonales. Al desplazar el ratón sobre esta zona, aparece un cursor en forma de cruz. Al hacer clic y arrastrarlo se logrará dividir y combinar las ventanas. Esta opción permite, como se ve en la Figura T1.13, configurar el espacio de trabajo como le sea más cómodo.

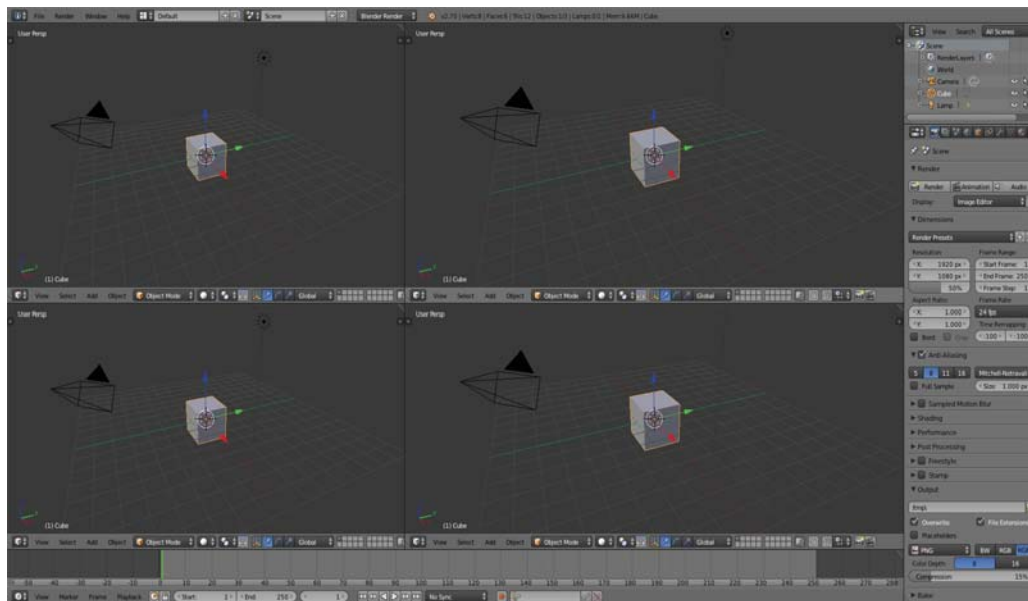


Figura T1.13. Ejemplo de configuración del espacio de trabajo

2.4 La vista 3D

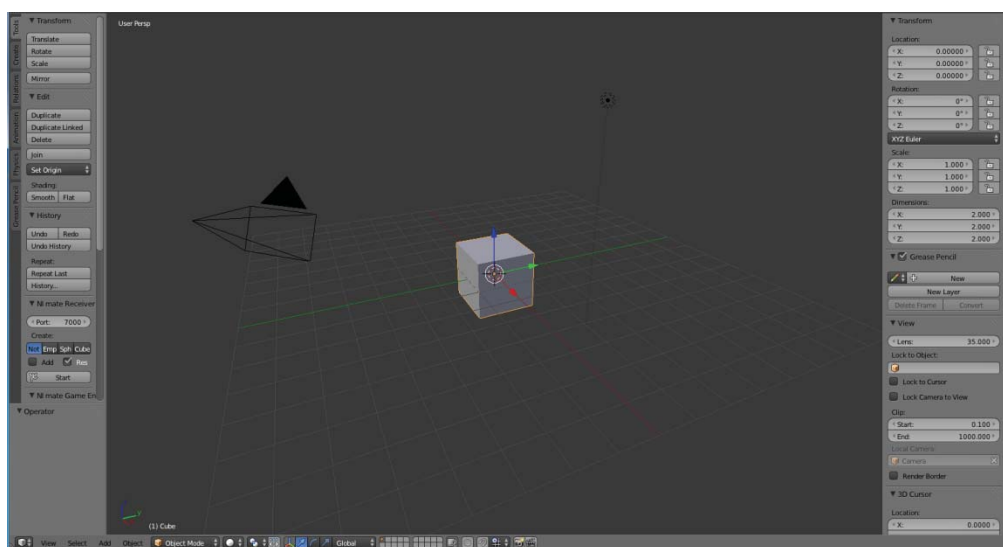


Figura T1.14. Vista 3D por defecto de Blender

Este es el espacio de trabajo (Figura T1.14) donde se verán los modelos y el que más se usará a lo largo de este tutorial. En él, se encontrará la ventana principal, que, por defecto, se abre con un cubo, un foco de luz predeterminado y una cámara, los cuales se pueden borrar para iniciar el proyecto.

La columna de la izquierda es la barra de herramientas. Se puede hacer aparecer o desaparecer pulsando la tecla T. La columna de la derecha es la barra de propiedades, la cual se hará aparecer o desaparecer con la tecla N.

2.5 Línea de tiempo o Timeline

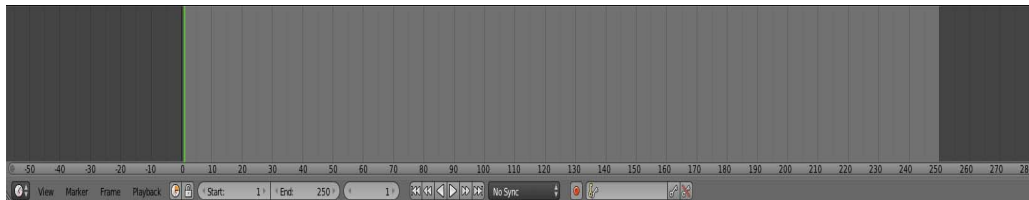


Figura T1.15. Timeline de Blender

Para poder manipular el tiempo de las escenas de animación se necesita la herramienta de línea de tiempo, o Timeline (Figura T1.15). En el Timeline, se dispone de los botones necesarios para recorrer la animación en el tiempo, así como las herramientas necesarias para crear la escena animada.

2.6 Ventana Outliner

En la esquina superior derecha de la pantalla se encuentra la ventana Outliner (Figura T1.16). Es una representación de la estructura de la escena. Todos los objetos, materiales, datos de animación y parámetros de escena se encuentran aquí. En una escena compleja con muchos objetos con capas, es mucho más sencillo buscar un objeto por su nombre que intentar localizarlo entre las líneas y vértices de la vista 3D.

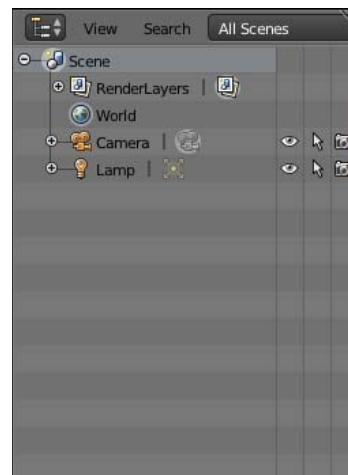


Figura T1.16. Ventana Outliner

2.7 El buscador de herramientas

Al pulsar la barra espaciadora del teclado en Blender, se abre el buscador de herramientas (Figura T1.17). Es una función de búsqueda interactiva. Poniendo la palabra

clave de la herramienta que se necesita, automáticamente sale un listado con las posibles herramientas que contenga esa palabra (siempre en inglés). Además, muestra el método abreviado de teclado de la opción.

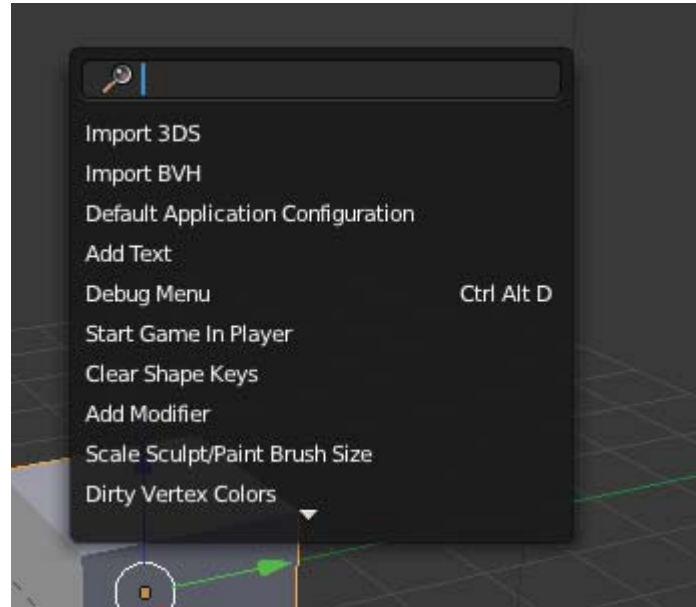


Figura T1.17. Buscador de herramientas de Blender

Tutorial Parte 2.

Animación con datos de captura de movimiento externos

En esta parte del tutorial, se realizará, la captura de movimiento, a partir de datos de captura de movimiento externos, es decir, grabados y exportados, animar cualquier modelo.

2.1 Animación con datos mocap externos (.BVH)

Lo primero que se hará, será abrir el archivo “modelohumano” de la carpeta “modelos”. Y aparecerá una ventana como la que se muestra a continuación.

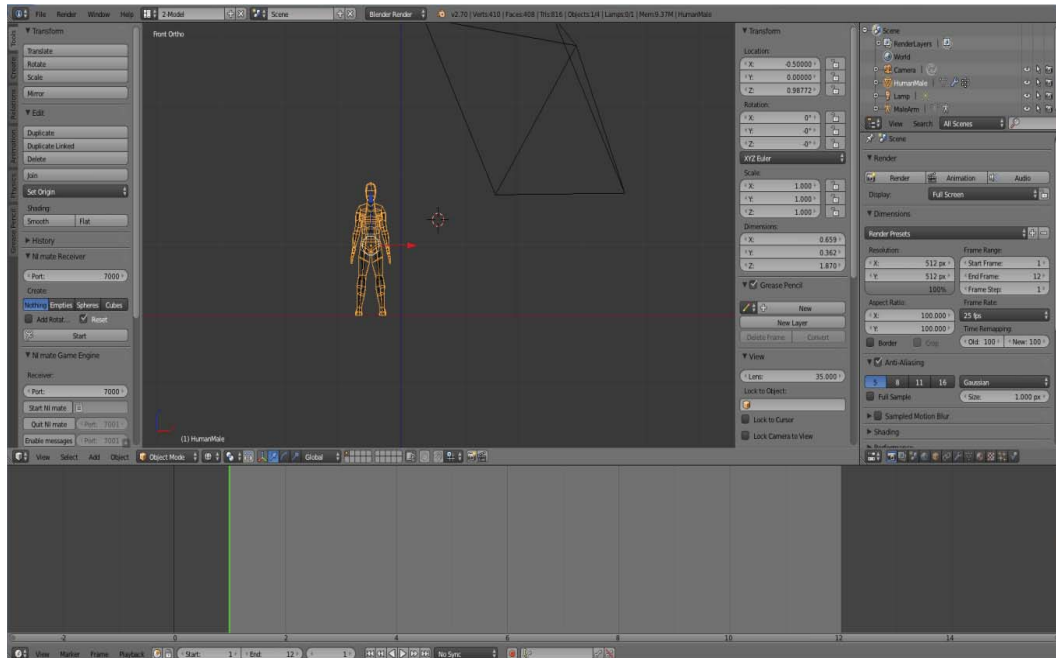


Figura T2.1. Pantalla al iniciar el proyecto “modelohumano”

A continuación se comprobará que el modelo tiene esqueleto. Esto se hace haciendo clic con el botón de seleccionar (botón derecho del ratón) en el centro de nuestro personaje.

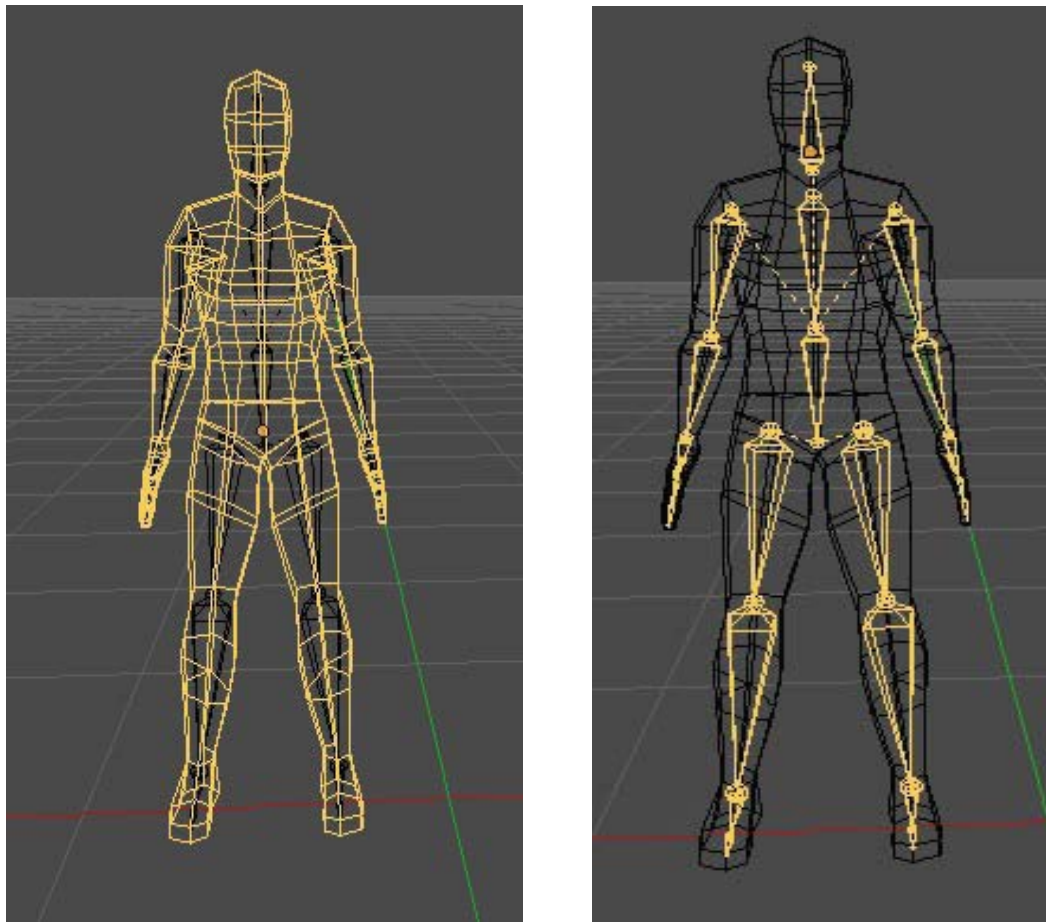


Figura T2.2. A la izquierda, el modelo que la piel seleccionada. A la derecha, el modelo con el esqueleto seleccionado

Como se observa en la Figura anterior, nuestro personaje se divide en dos partes claramente visibles: la piel o skin (imagen de la izquierda) y el esqueleto (imagen de la derecha).

Para añadirle datos externos de captura de movimiento a nuestro modelo, necesitamos conocer el nombre de cada hueso. Para ello se selecciona el esqueleto, y, en la columna de la derecha pulsamos sobre el icono que se ve en la Figura T2.3 y marcamos el cuadro “names”.

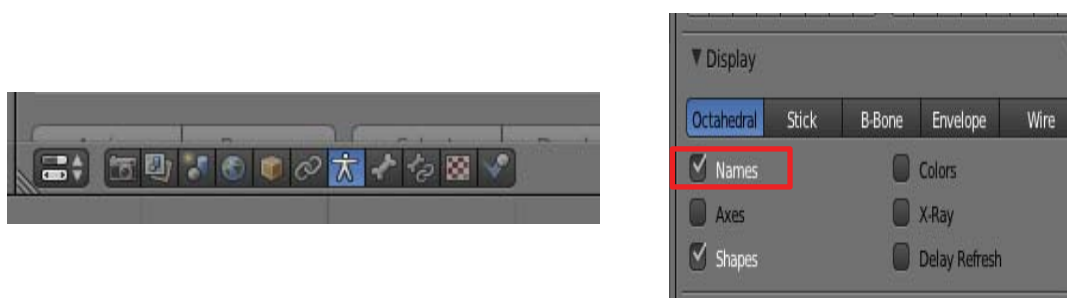


Figura T2.3. A la izquierda el icono “armature” seleccionado. A la derecha, la casilla que se debe marcar para visualizar los nombres de los huesos.

Al hacer esto, se comprueba que el esqueleto queda de la siguiente forma:

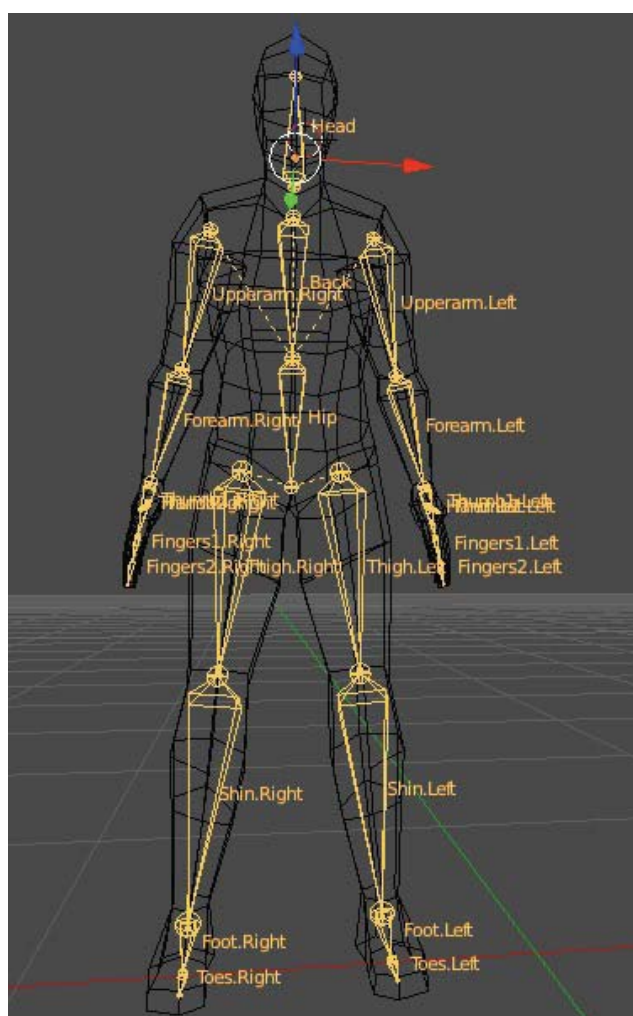


Figura T2.4. Esqueleto con los nombres de los huesos visibles.

Una vez hecho esto, lo siguiente será importar de la carpeta “DatosBVH” el movimiento “caminar”. Para ello, se pulsará File→Import→Motion Capture (.bvh), como se muestra en la siguiente figura (Figura T2.5):

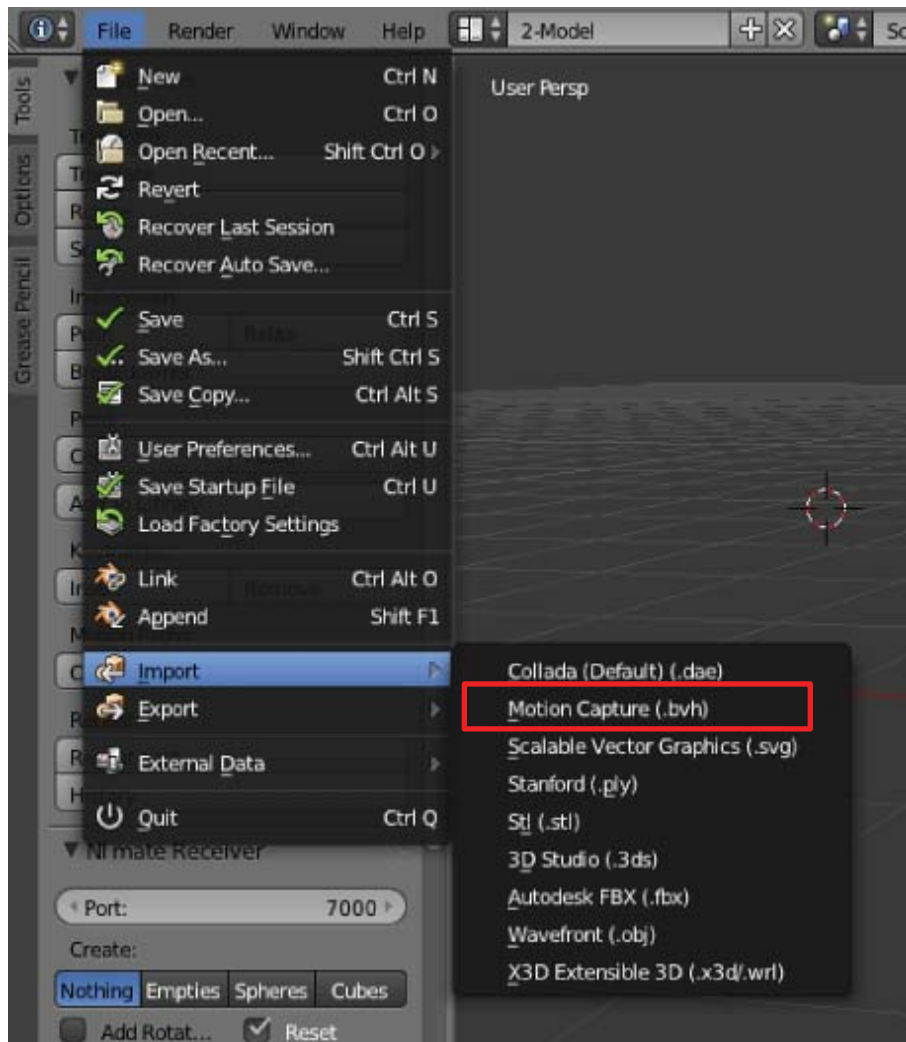


Figura T2.5. Proceso para importar un archivo externo de datos MOCAP.

Se busca el archivo “caminar.bvh” y se pulsa en “import BVH” (arriba a la derecha) como se muestra en la Figura T2.6.

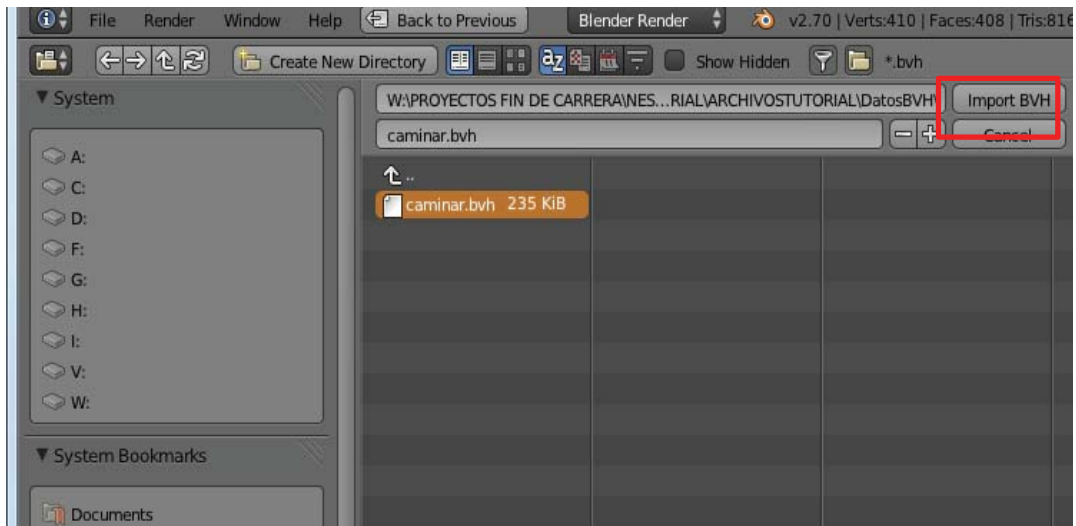


Figura T2.6. Pantalla que aparece para importar los datos BVH.

Volverá a la vista principal, donde se deberá alejar el zoom para ver un esqueleto nuevo y muy grande. Se deberá seleccionar y pulsar la tecla “S” para reducir su tamaño, moviendo el ratón hasta tener un tamaño parecido al de nuestro modelo. Con la tecla “G” se podrá mover para tenerlo cerca del modelo y sea más cómodo para trabajar.

Para ver como se mueve, se pulsará el play en el timeline o “Alt+A”, y se reproducirá el movimiento de los datos externos. Se deberá tener algo como lo siguiente:

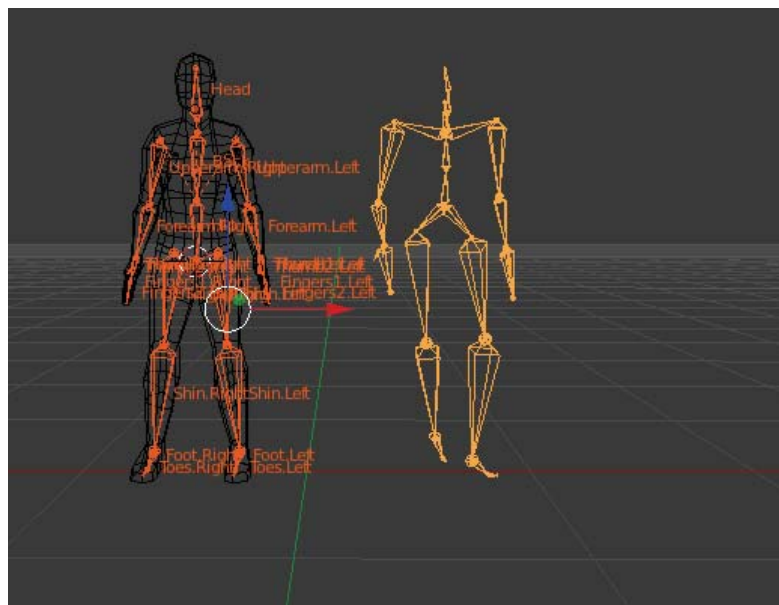


Figura T2.7. A la izquierda, el modelo “modelohumano”. A la derecha, el esqueleto con los datos externos de MOCAP.

Al igual que se hizo antes, seleccionamos el nuevo esqueleto y seleccionamos la casilla de “name” y obtendremos lo siguiente.

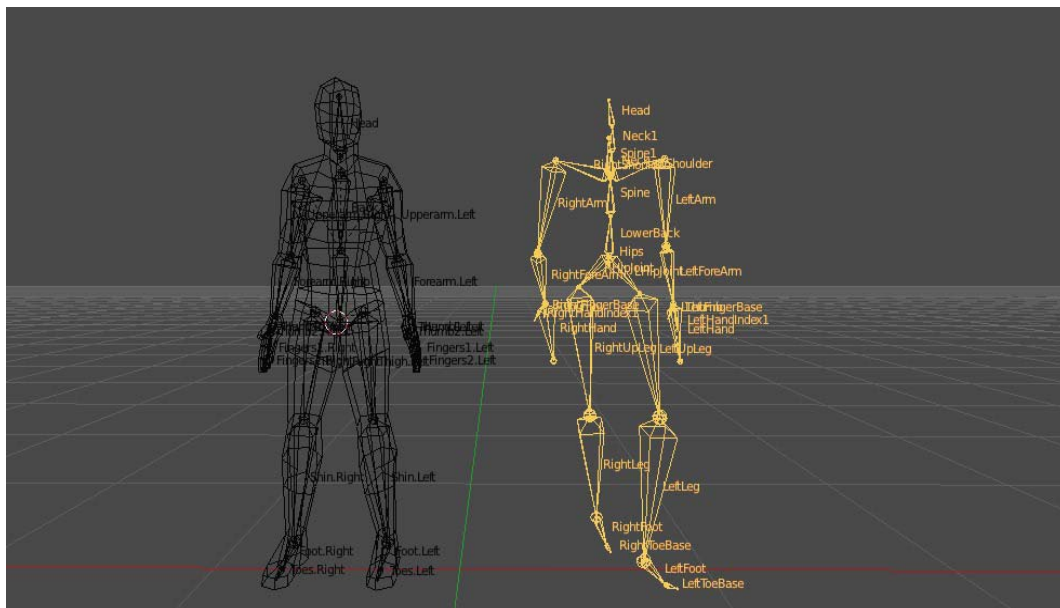


Figura T2.8. Esqueleto con datos MOCAP con los nombres de los huesos visibles.

Se seleccionará el modo postura, “pose mode”, y seleccionamos el esqueleto de nuestro modelo. Ahora en el encabezado de la columna de la derecha aparecerá un nuevo icono, que es un hueso con dos anillas (Figura T2.9). Aquí es donde se añade restricciones (constraint) a los huesos del modelo. Las dos restricciones que se van a utilizar son “copy rotation” y “copy location”.

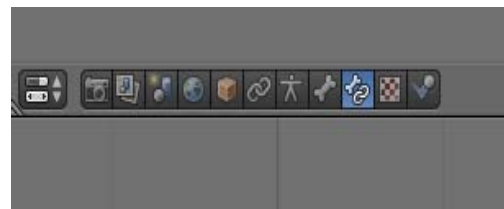


Figura T2.9. Lugar donde se añaden las restricciones a los huesos del modelo.

Se comenzará por seleccionar el hueso de la cabeza del modelo, a continuación hacemos clic en “Add Bone Constraint” y en “copy rotation” y aparecerá una ventana, justo debajo. En ella se pulsará en la barra en blanco llamada target (objetivo, en español) y se seleccionará el nombre de los datos externos de movimiento (“caminar”). Con ello, aparecerá una nueva barra, en la que se seleccionará el hueso al que queremos copiar el movimiento. En este caso, el hueso de la cabeza (“head”), como se muestra en la Figura T2.10. Al finalizar, se verá un cambio de posición en la cabeza de nuestro modelo, muy similar al de los datos externos de movimiento. Lo que se ha hecho es copiar el movimiento de los datos externos de ese hueso al nuestro.

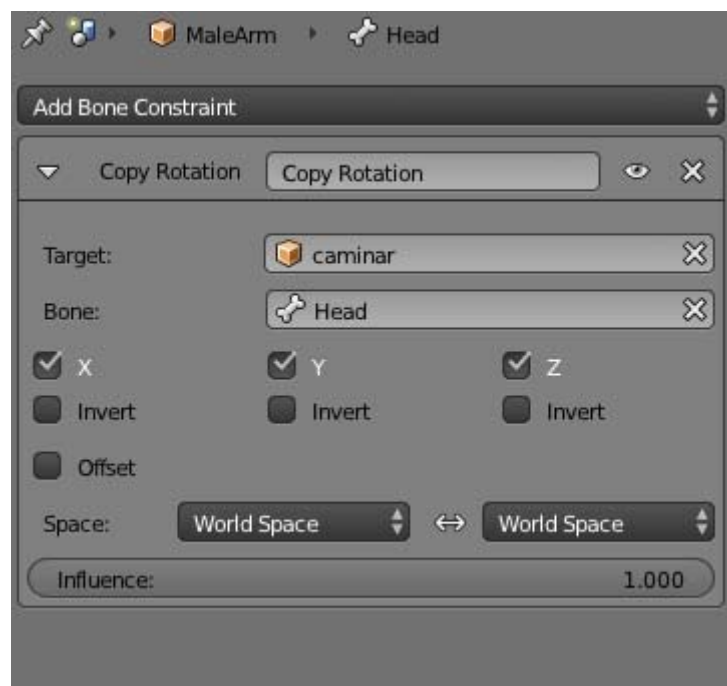


Figura T2.10. Ejemplo de cómo añadir una restricción al hueso de la cabeza del modelo.

Se repetirá el mismo proceso para todos los huesos del modelo. A continuación se muestra una figura, donde se movimiento copia cada hueso.

Modelo	Datos Externos
Head	Head
*Back	*Spine
*Hip	*Hips
Upperarm.Right	RightArm
Forearm.Right	RightForeArm
Hand.Right	RightHand
Upperarm.Left	LeftArm
Forearm.Left	LeftForeArm
Hand.Left	LeftHand
Thigh.Right	RightUpLeg
Shin.Right	RightLeg
Foot.Right	RightFoot
Toes.Right	RightToeBase
Thigh.Left	LeftUpLeg
Shin.Left	LeftLeg
Foot.Left	LeftFoot
Toes.Left	LeftToeBase

Figura T2.11. Proceso para añadir restricciones a cada hueso del modelo.

*Al copiar estos dos movimientos, como se habrá comprobado, el modelo se dobla. Esto es debido a que está invertido el hueso. Para solucionarlo, en la ventana marcaremos debajo del eje X, la casilla “invert” para invertirlo, como se muestra a continuación.

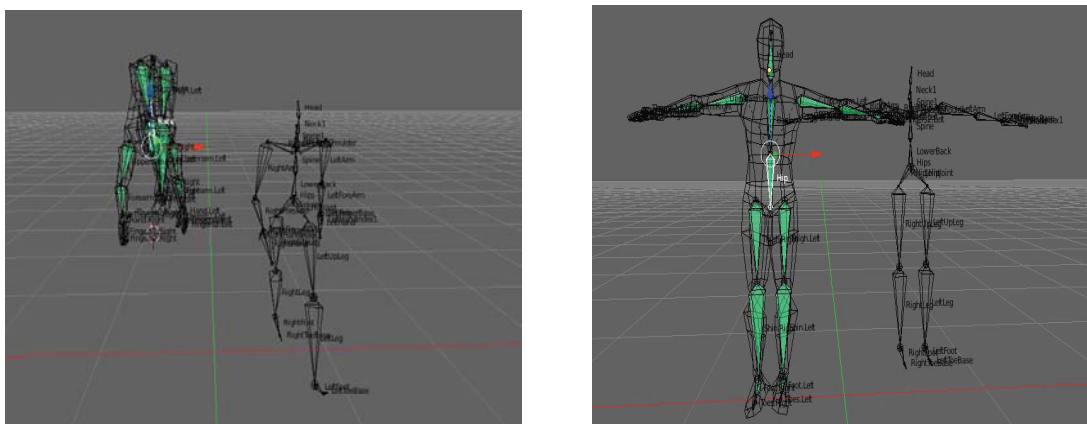


Figura T2.12. A la izquierda, ejemplo del modelo sin marcar la casilla “invert” del eje X del torso. A la derecha, ejemplo del modelo con las casillas “invert” del eje X del torso marcadas.

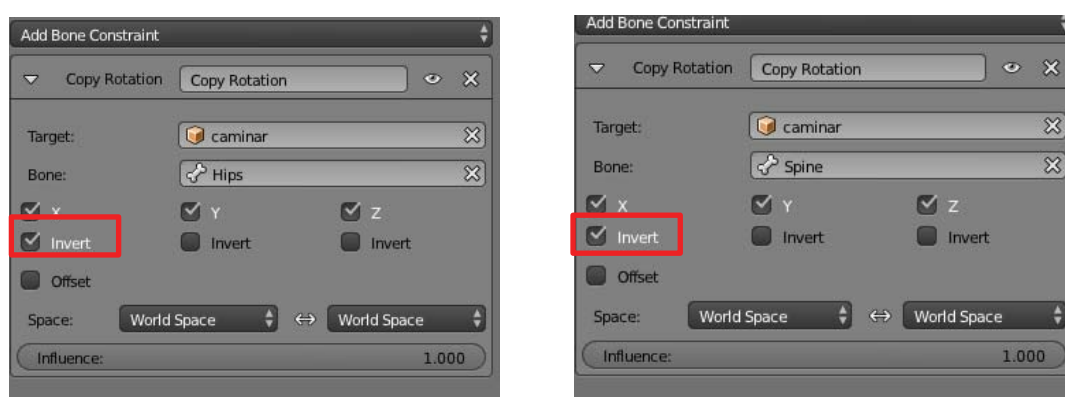


Figura T2.13. Muestra de los dos huesos del torso del modelo con la casilla “invert” marcada.

Al reproducir de nuevo, se verá como el modelo copia el movimiento de los huesos de los datos externos, pero no se desplaza como él. Para solucionar esto, se tendrá que añadir al hueso central del modelo la otra restricción que se comentó con anterioridad, es decir, “copy location”.

Se selecciona el hueso “Back” del modelo y se hace clic en “Add Bone Constraint”, pero esta vez, se pulsará “copy location”. Aparecerá otra ventana, justo debajo de la anterior restricción y se seleccionará el hueso correspondiente (Spine) de los datos externos (Figura T2.14).

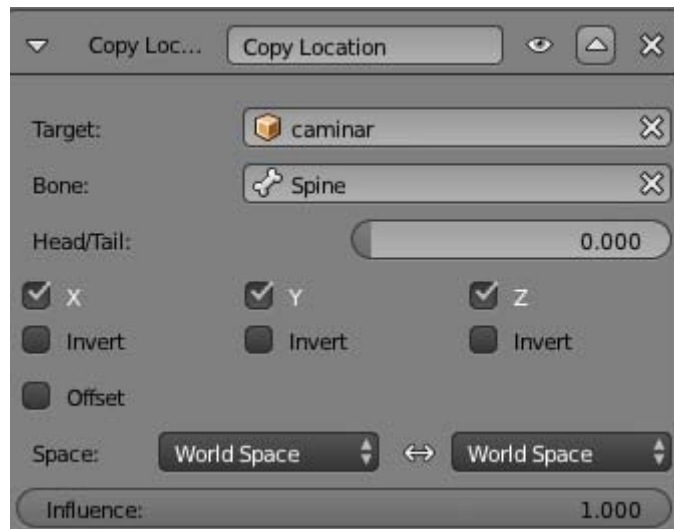


Figura T2.14. Muestra del proceso “Copy Location”.

Se obtendrá algo parecido a lo que se muestra a continuación:

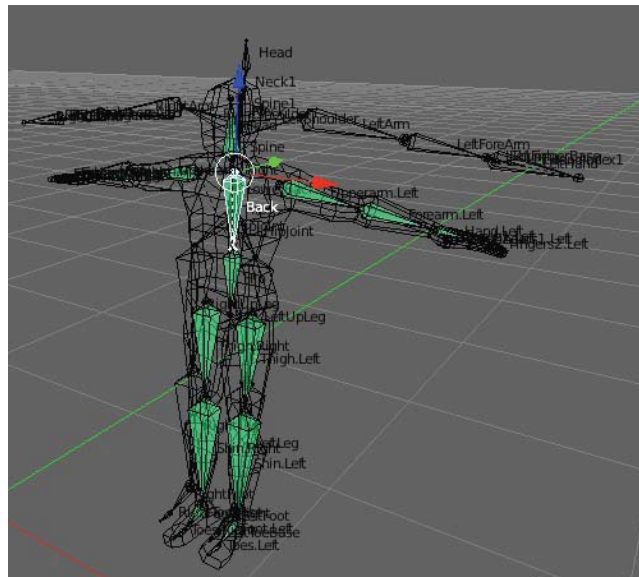


Figura T2.15. Muestra de lo que sucede al añadir la restricción de copia de la ubicación (“copy Location”).

El modelo se ha puesto en la misma posición que los datos externos, como se observa, es un tanto engorroso. Para que sea más cómodo a la hora de trabajar con él, pulsaremos en la casilla “offset” (Figura T2.16) y pulsando la tecla “G” podremos separarlo para tener una vista de los dos por separado (Figura T2.17).

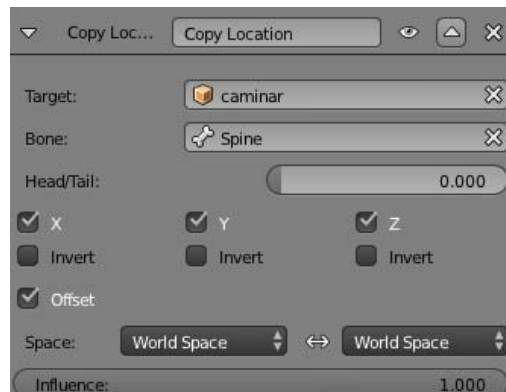


Figura T2.16. Muestra de “Copy Location” con la casilla “Offset” marcada.

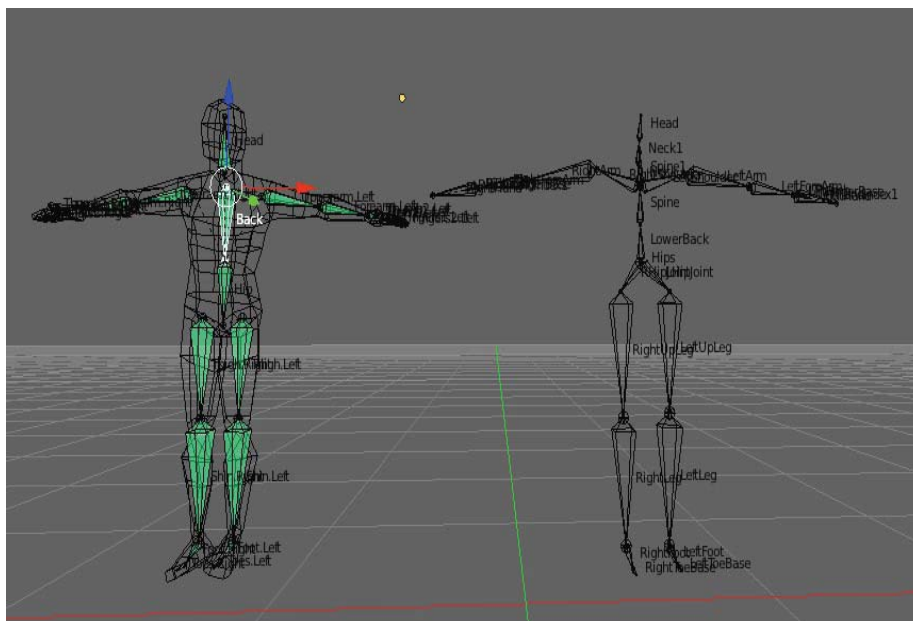


Figura T2.17. Muestra de nuestro modelo desplazado a la izquierda para mayor comodidad a la hora de trabajar con él.

Al pulsar reproducir (“Alt+A”) se comprobará que el modelo ya se desplaza.

2.2 Corrección de detalles

Existen algunos detalles, como el de los pies, que están torcidos, que se podrá corregir marcando la casilla “offset” del hueso, en este caso el de los pies derecho e izquierdo, y rotarlo al gusto de cada uno con la tecla “R”. Lo mismo para cualquier otra parte del cuerpo que se quiera modificar.

Tutorial Parte 3.

Realización de Captura de Movimiento en Tiempo Real

En esta parte del tutorial se realizará la captura de movimiento en tiempo real, comenzando con la configuración de los softwares que se usarán. A continuación, se crearán una serie de proyectos en Blender con diferentes formas de llegar a la captura de movimiento en tiempo real.

3.1 Configuración del software

En la primera parte de este tutorial, se explicó el procedimiento para descargar e instalar los software que se van a usar. Para la realización de una captura de movimiento en tiempo real, se deben seguir unos pasos previos.

3.1.1 Configuración de Blender

Se comenzará con la configuración del software Blender. Para ello, se deberá situar en la página principal de Blender y pinchar arriba a la izquierda: “File→User Preferences...” (Figura T3.1, a la izquierda) y aparecerá una ventana en la que se deberá pulsar “Addons” como se muestra en la figura, a la derecha.

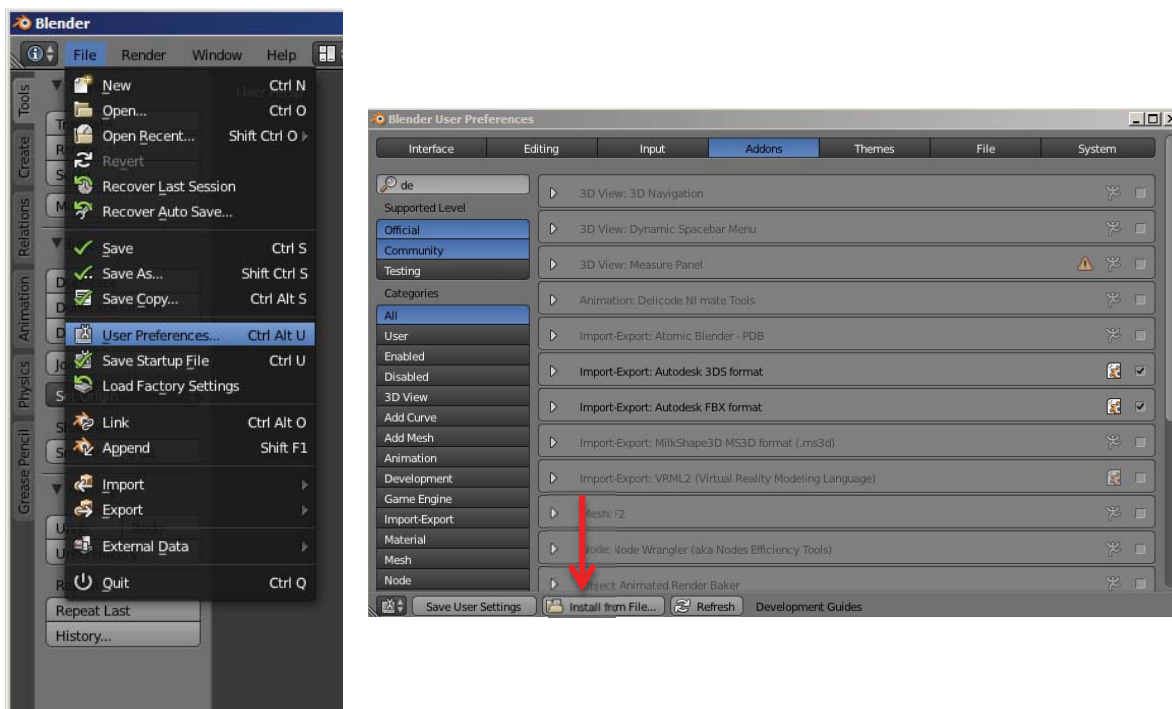


Figura T3.1. A la izquierda, como llegar a “User Preferences...”. A la derecha, la ventana “Addons”.

Una vez allí, en la parte de debajo de dicha ventana, se deberá pulsar “Install From File...” que aparece señalado con una flecha en la Figura anterior (Figura T3.1, a la derecha). A continuación, deberá buscar el archivo “animation_delicode_nimate_tools.py” que está incluido en la carpeta “configuración→ Addon_Blender” de los archivos adjuntados a este tutorial. Al seleccionarlo, se deberá pulsar “Install From File...” situado arriba a la derecha.

Al hacer lo anterior, aparecerá la ventana “User Preferences..” con el addon del software Delicode Ni-Mate el cual se deberá marcar para poder usarlo como se muestra en la Figura T3.2.



Figura T3.2. Pestaña que se debe marcar para poder usar las herramientas del software Delicode Ni-Mate.

Una vez hecho esto, en la barra de herramientas de Blender, columna de la izquierda, se podrán observar unas herramientas nuevas (Figura T3.3). A lo largo de este tutorial, se irán usando las necesarias para la realización de la captura de movimiento en tiempo real.

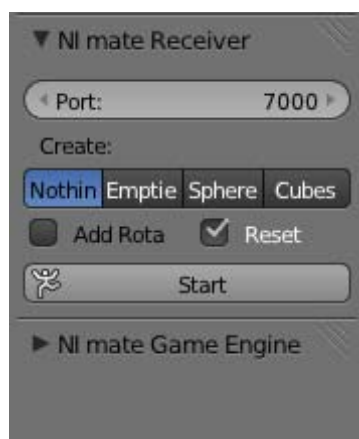


Figura T3.3. Ventana de herramientas del software Delicode Ni-Mate.

3.1.2 Configuración de Delicode Ni-Mate

Al ejecutar el software (sólo se ejecutará si está conectada la Kinect) aparecerá una ventana como la que se muestra en la Figura T3.4. Lo que se debe hacer es cargarle una configuración predeterminada para usar el kinect con el Blender. Para ello, habrá que importar el archivo de configuración “simplebunny.config”, el cual se encuentra en la carpeta Tutorial/Configuración/Configuración_Delicode.

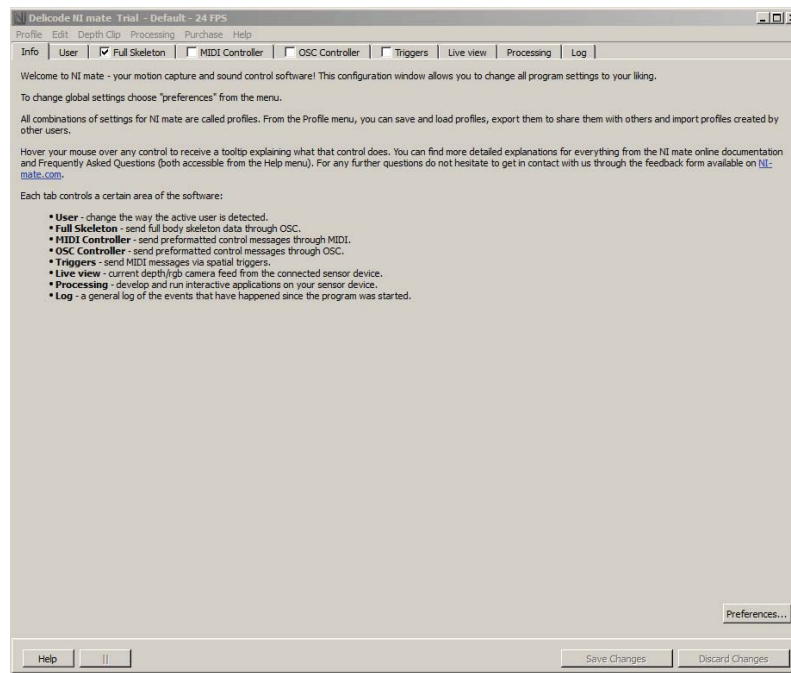


Figura T3.4. Ventana principal del software Delicodex Ni-Mate.

Una vez importada, habrá que cargarla dentro del software. Para ello, pinchamos en “File→Load→ BunnySimple”, como se muestra en la Figura T3.5, y se cargará automáticamente en el software.

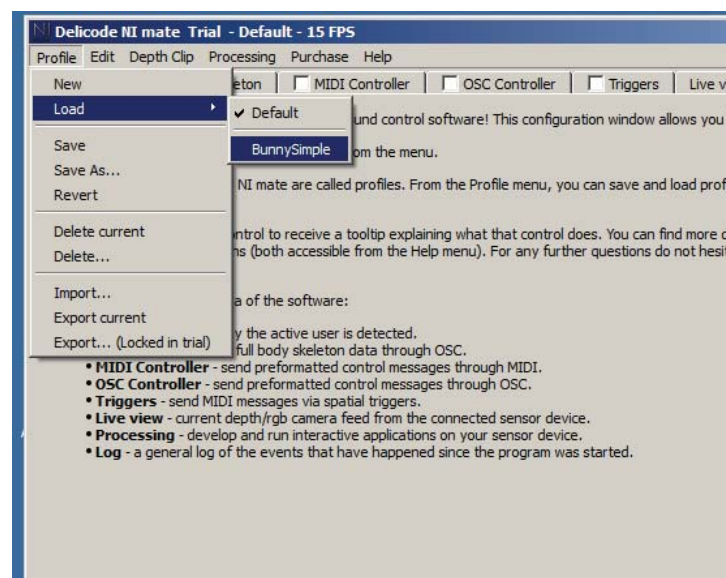


Figura 3.5. Carga del archivo “BunnySimple”.

Una vez cargado el archivo de configuración, deberá pulsar en “Full Skeleton” y se mostrará una ventana, como la que se muestra a continuación, en la que se marcará varias cosas a tener en cuenta:

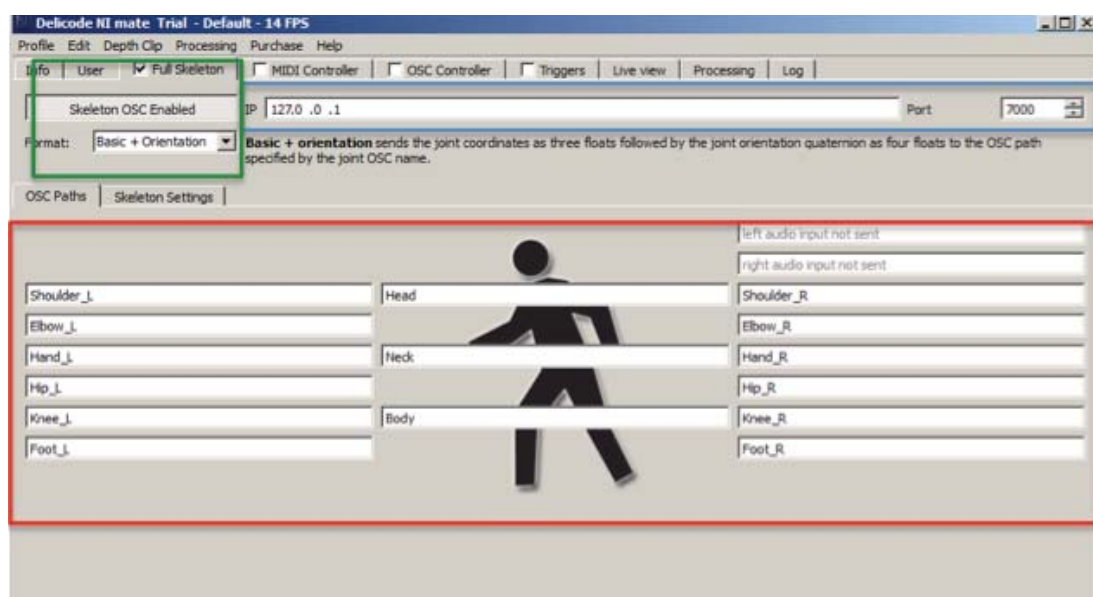


Figura 3.6. Datos importantes a tener en cuenta. En rojo, los nombres de los huesos que se deben poner en Blender. En azul, la dirección IP y el número del puerto que deben coincidir con los que aparecen en la barra de herramientas del software Delicodelite Ni-Mate en la columna izquierda del Blender. En verde, selección del formato “Basic+Orientation” y marcar “Full Skeleton”.

La primera, marcada en rojo, son los nombres de los huesos. Esto es muy importante, ya que, si no se escriben exactamente igual (en Blender) que salen aquí en los pasos siguientes, no podrá realizar la captura de movimiento.

Lo segundo que tendrá que tener en cuenta es el puerto y la IP, marcada en azul. Si no están estos datos en Blender, no se comunicará un software con otro y, por consiguiente, no podrá realizar la captura de movimiento.

Por último, se deberá asegurar que en la pestaña, marcada en verde, está la opción que se muestra, es decir, “Basic+ Orientation”. También, tendrá que comprobar que está marcada la casilla de “Full Skeleton”.

Con esto se finaliza la configuración de los softwares. A continuación se comenzará a crear el proyecto en Blender.

3.2 Creación del proyecto de captura de movimiento en Blender

Se deberá situar en la pantalla principal de Blender, con el Delicode Ni-mate encendido, y la Kinect conectada, colocada a una altura de 1 metro aproximadamente. A continuación se realizará una prueba para comprobar que todo funciona correctamente.

En la columna de la izquierda, donde están, ahora las herramientas del Delicode Ni-mate, se pulsará “empties” y, acto seguido, en “Start” (Figura 3.7, a la izquierda). Se deberá colocar delante de la Kinect en posición de calibración (piernas abiertas y brazos en cruz), a una distancia de 1,80 metros aproximadamente. Y lo que se mostrará en la vista principal es una especie de esqueleto humano, que en vez de huesos serán ejes, como el que se muestra en la Figura 3.7, a la derecha.

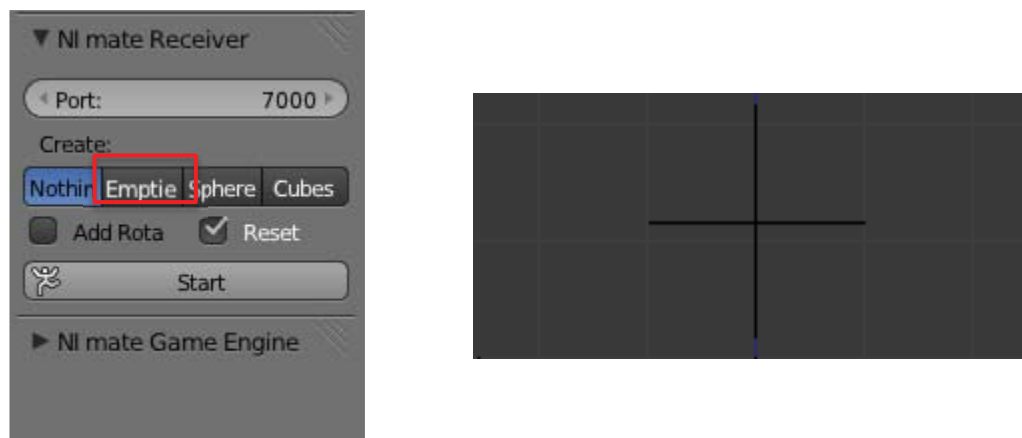


Figura 3.7. A la izquierda, la opción “Empties” marcada en rojo. A la derecha, ejemplo de un eje creado.

Si por algún casual no aparece en el centro de la pantalla, se deberá colocar la vista con ayuda del ratón. Muchas veces no aparecerá centrado. Una vez esté bien posicionado, ya se podrá hacer las pruebas de los movimientos que se quiera.

Una vez comprobado que funciona, para comprender mejor lo que se está haciendo se va a realizar paso a paso la construcción y configuración de los ejes.

3.3 Realización de captura de movimiento con un esqueleto creado paso a paso.

Se abrirá un proyecto nuevo en Blender, y se eliminarán los tres objetos que se muestran por defecto, es decir, el cubo, la cámara y el foco de luz. Una vez eliminados, se cambiará la vista a vista frontal (Figura T3.8) pulsando 1 en el “numb pad” o , manualmente, pulsando “view→front” situado en el encabezado de la vista 3D.



Figura T3.8. Vista Frontal.

Donde cortan los ejes se añadirá el primer eje. Para añadirlo, se pulsará justo en el centro con el botón izquierdo del ratón y, a continuación, se deberá pulsar en “add→Empty→Plain Axes” (Figura T3.9, a la izquierda). Automáticamente se añadirá un eje en la posición que se tenía marcada.

Si se selecciona ese eje y se accede, en el menú de propiedades, a las propiedades del objeto (el cubo naranja), lo primero que aparece, justo debajo, es el nombre del eje, el cual, por defecto, será “Empty”. Se deberá cambiar el nombre y para ello, pulsamos en la barra donde aparece y lo cambiamos por “Neck” (uno de los nombres que aparecía en la pantalla “full skeleton” del software Delicode Ni-Mate), y pulsamos Enter (Figura T3.9, a la derecha).

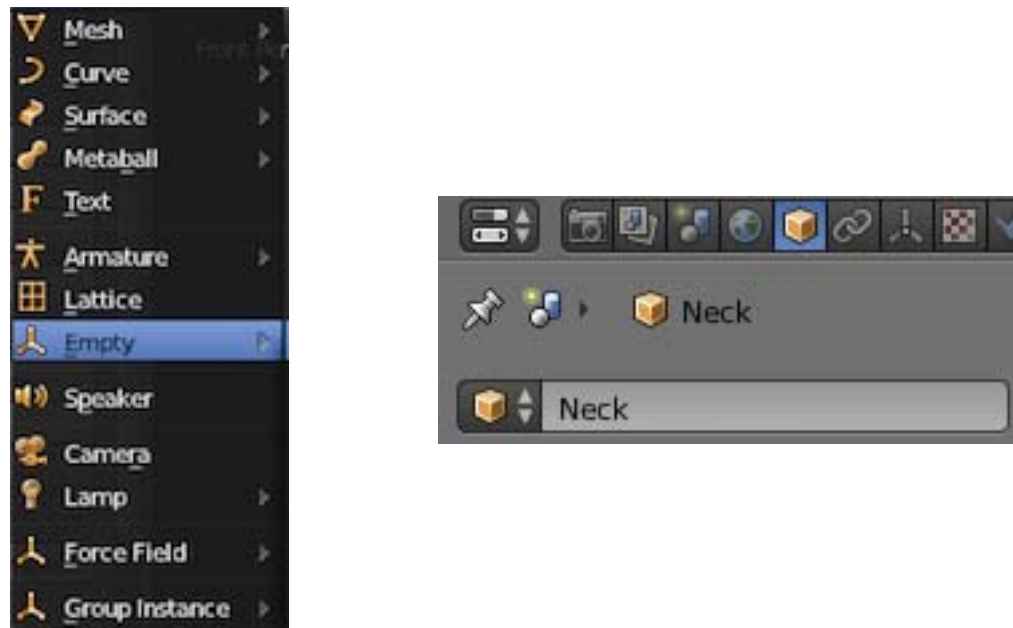


Figura T3.9. A la izquierda, lugar donde se encuentran los ejes. A la derecha, propiedades del objeto y el nombre modificado.

A continuación, seguiremos añadiendo ejes de la misma forma que el anterior, pero colocándolos de manera que se asimile a una forma humana con la posición de calibración. En la siguiente figura (Figura T3.10) se muestra como debe quedar finalmente.

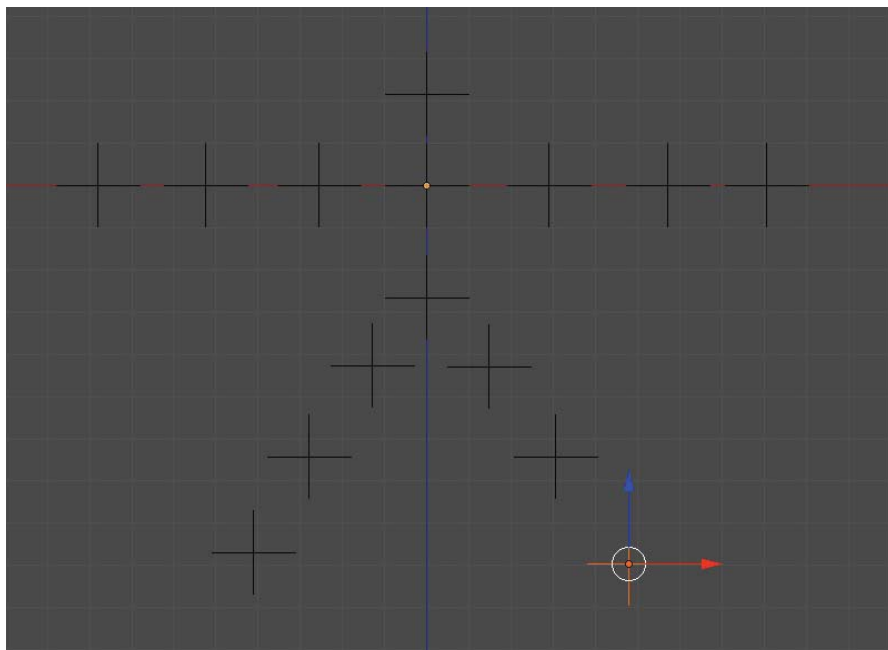


Figura T3.10. Ejemplo del resultado al colocar los ejes en disposición similar a un esqueleto humano.

Para nombrar cada eje de manera correcta, se señala en la Figura T3.11 y se añadirán siguiendo los mismos pasos anteriores. Tendrá que asegurarse de escribirlos exactamente como se muestra, ya que si, por ejemplo se olvida de una mayúscula, ya no funcionará correctamente. Si se recuerda, la parte de configuración de software, una de las partes importantes era el nombre de los huesos. Aquí se podrá comprobar que son exactamente los mismos nombres que en la ventana del software Delicode Ni-Mate.

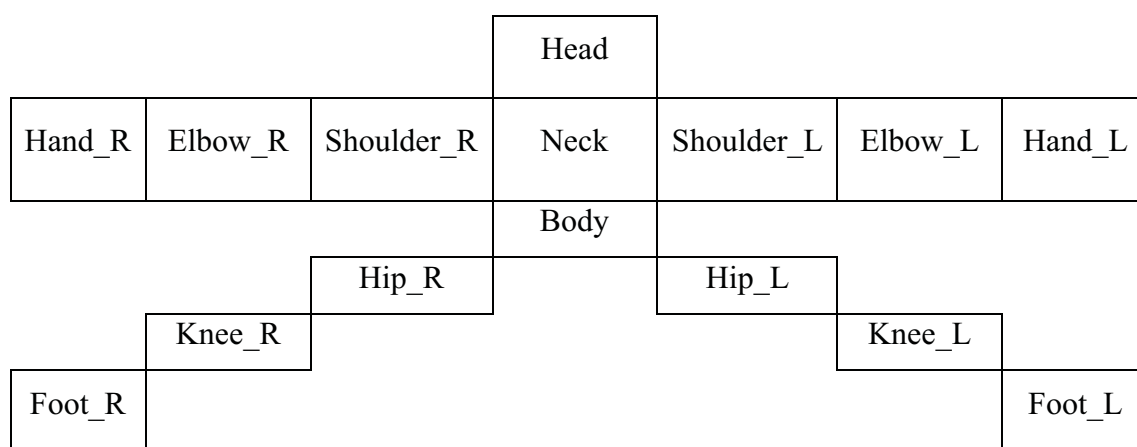


Figura T3.11. Cada recuadro representa a cada eje creado y los nombres que se le deben poner a cada uno.

Para comprobar que se mueve correctamente, se deberá pinchar, en la columna de la izquierda, en las herramientas del Delicode Ni-Mate, en “nothing” y seguidamente en “start”. Como siempre, se deberá calibrar la Kinect con la posición de calibración y, si se han seguido los pasos correctamente, esta estructura que se ha creado se moverá según la posición que se tenga.

A continuación se creará un esqueleto. Para ello deberá posicionarse en el eje al que ha nombrado “Body”, ya que será nuestro hueso raíz o principal, y pulsar en “add”→ “armature”→ “single Bone” (Figura T3.12) y aparecerá un hueso en el lugar marcado (Figura T3.13)

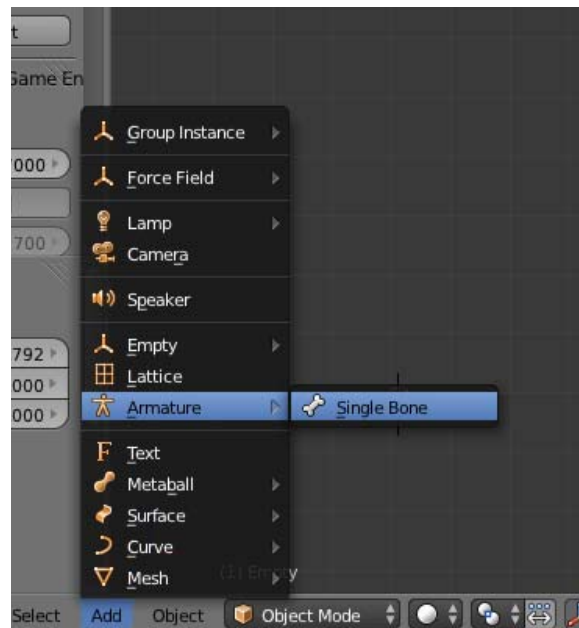


Figura T3.12. Añadir un hueso.

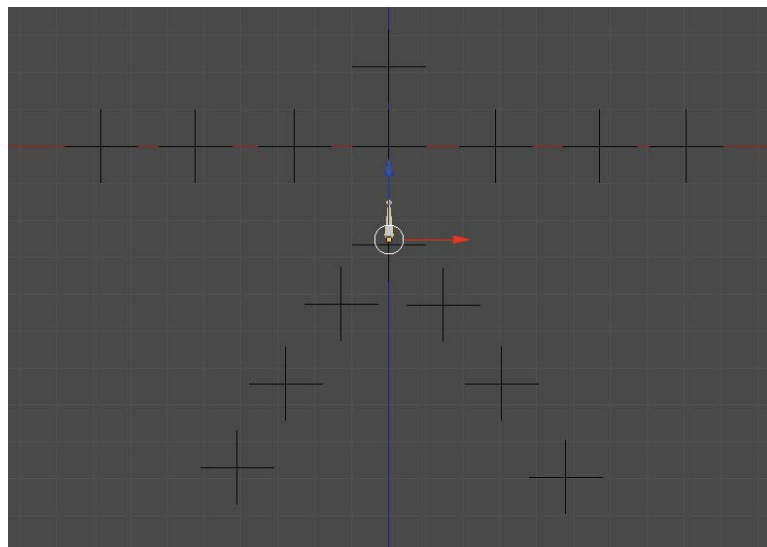


Figura T3.13. Hueso añadido en el lugar indicado con el ratón.

A continuación, se deberá cambiar a “Edit Mode” para seguir con la creación del esqueleto. Se pulsa en el extremo del hueso y se deberá pulsar la letra E para extruir el siguiente hueso, siempre siguiendo los ejes antes construidos. Es muy importante que si desea extruir un hueso que está por arriba, como el del cuello encima del cuerpo, se deberá hacer a partir del extremo más cercano.(Figura T3.14)

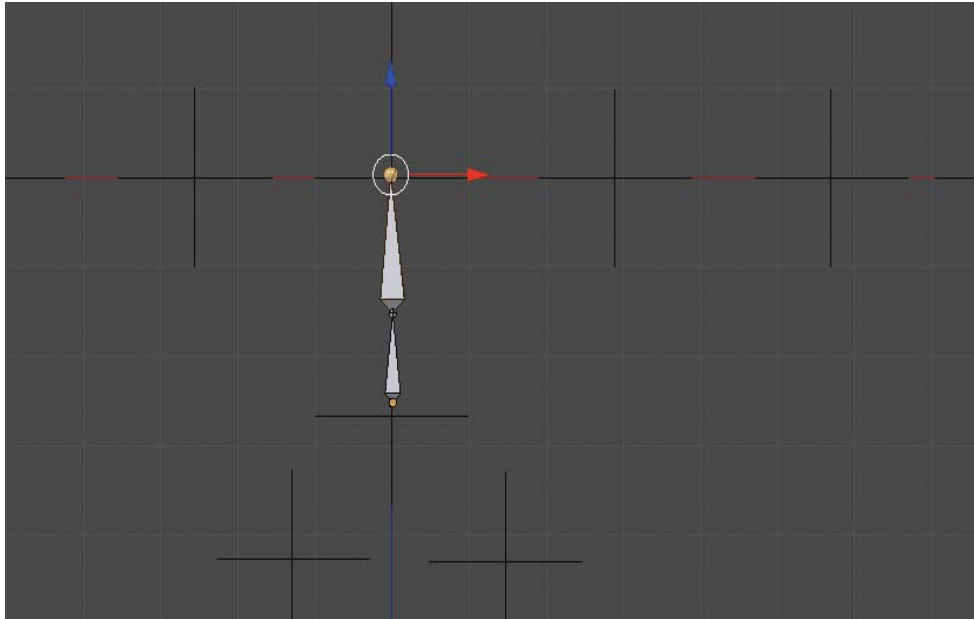


Figura T3.14. Ejemplo de extrusión de huesos.

Al finalizar de extruir cada hueso se tendrá algo similar a la siguiente figura:

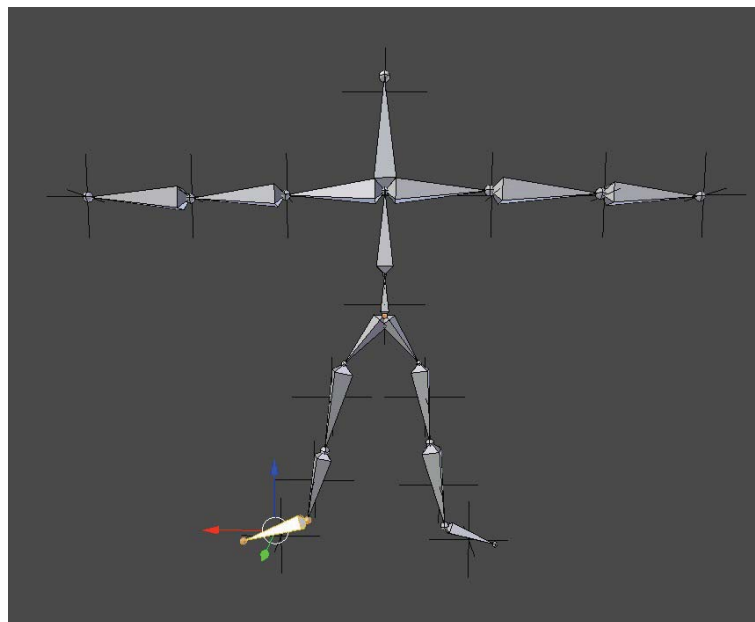


Figura T3.15. Ejemplo del esqueleto terminado

Como se puede observar en la Figura T3.15, se han colocado los extremos de los huesos aproximadamente en los ejes. También, se han añadido dos huesos más entre las piernas y el torso, las cuales darán más naturalidad en los movimientos.

Para que el esqueleto se mueva al igual que los ejes, deberá dirigirse a “pose mode” y en el menú de propiedades pinchar en restricciones de huesos, tal y como se hizo en la Parte 2 de este tutorial.

El paso que se describe a continuación es clave para que todo funcione correctamente. Se mostrará hueso a hueso, las restricciones que se deben poner.

- **Torso**

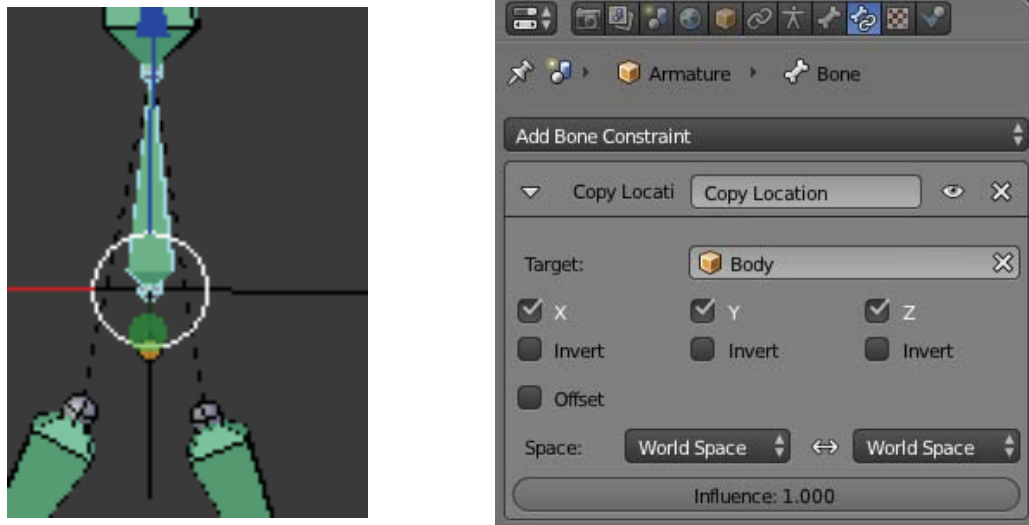


Figura T3.16. Hueso del torso y su restricción

- **Cuello**

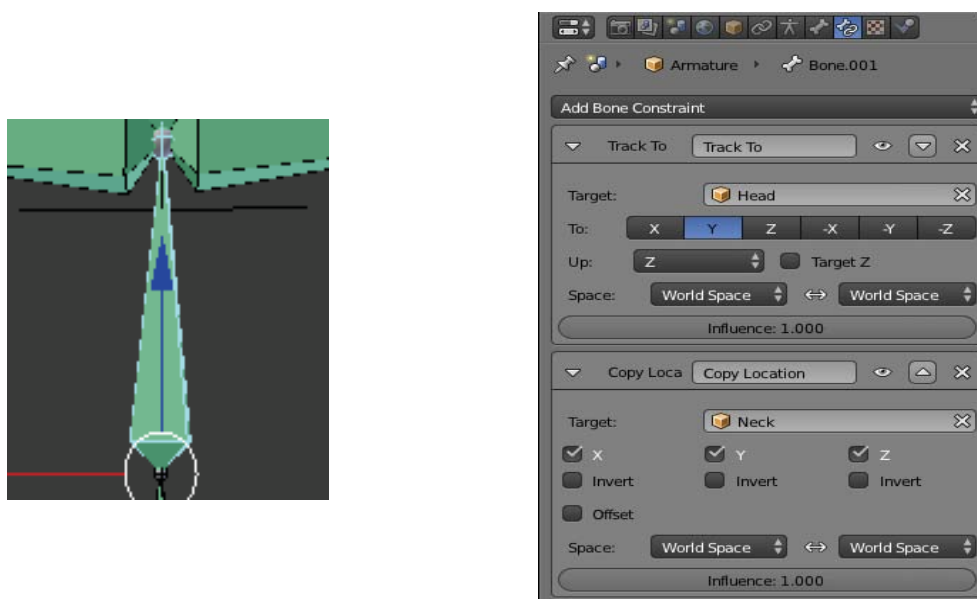


Figura T3.17. Hueso del cuello y sus restricciones.

- Cabeza**

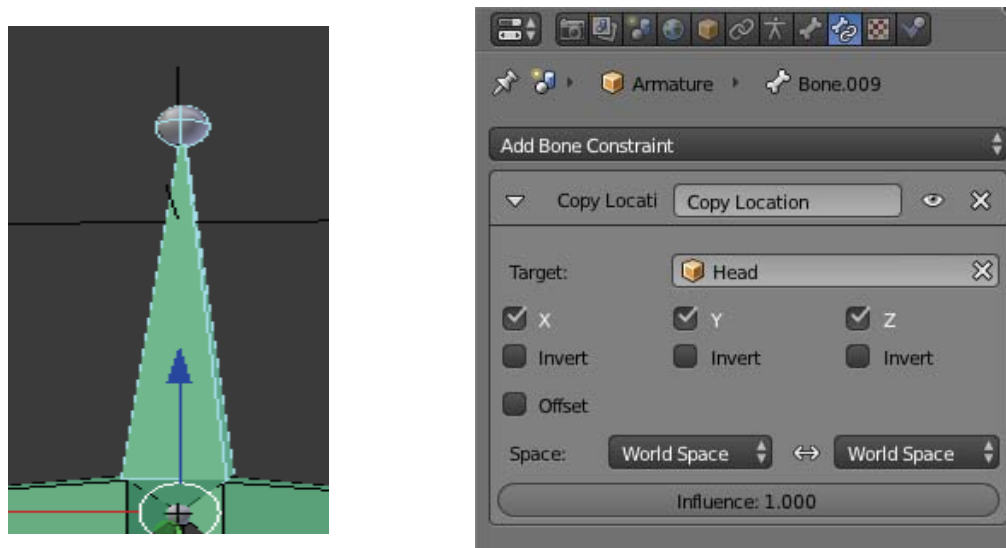
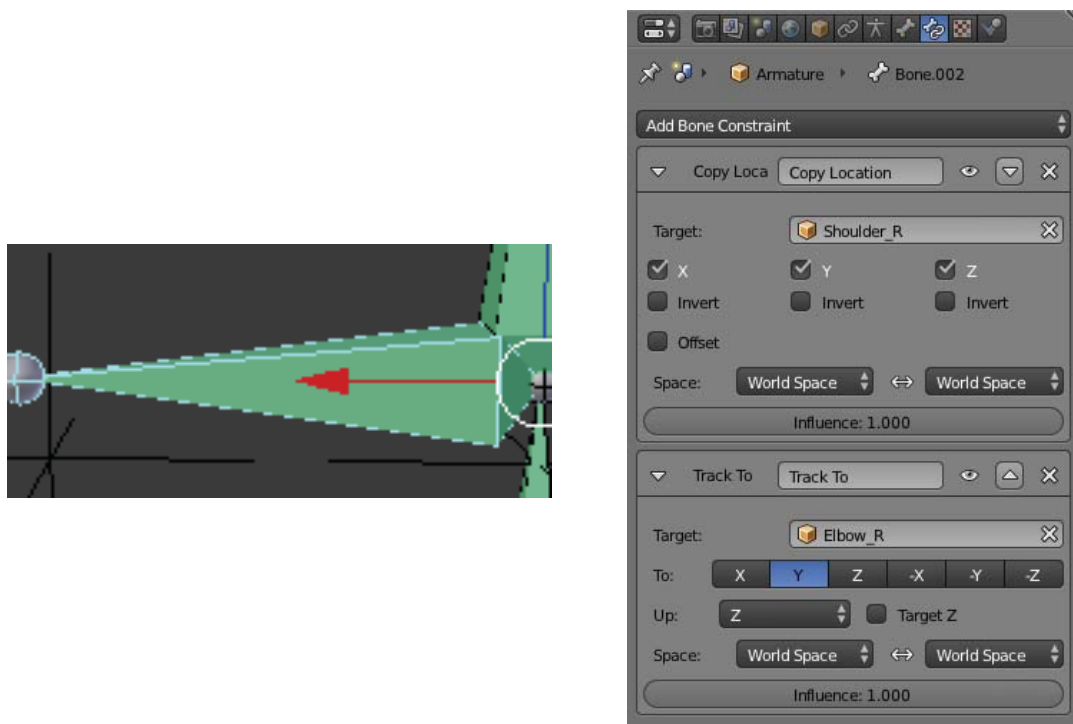


Figura T3.18. Hueso de la cabeza y su restricción.

- Hombro derecho, antebrazo derecho y mano derecha**



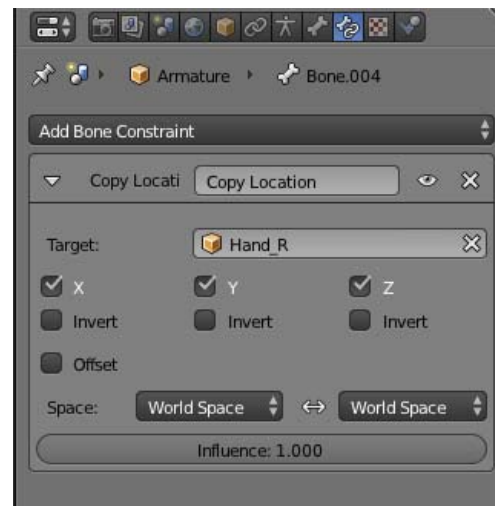
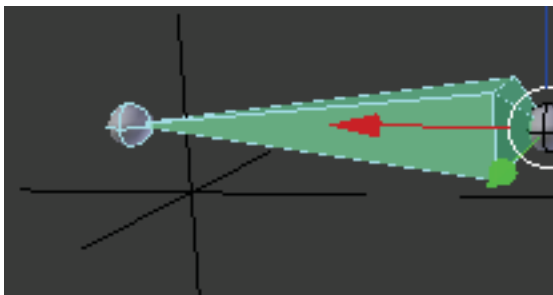
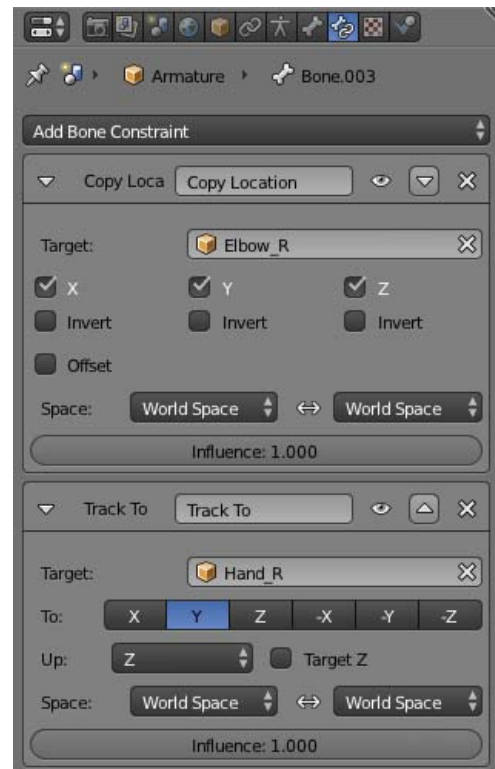
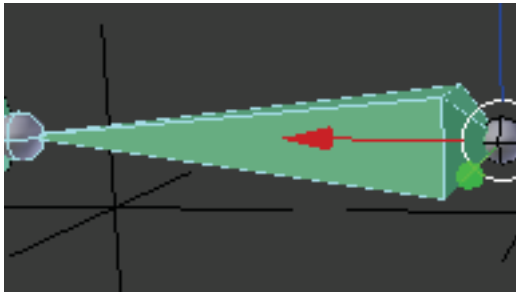
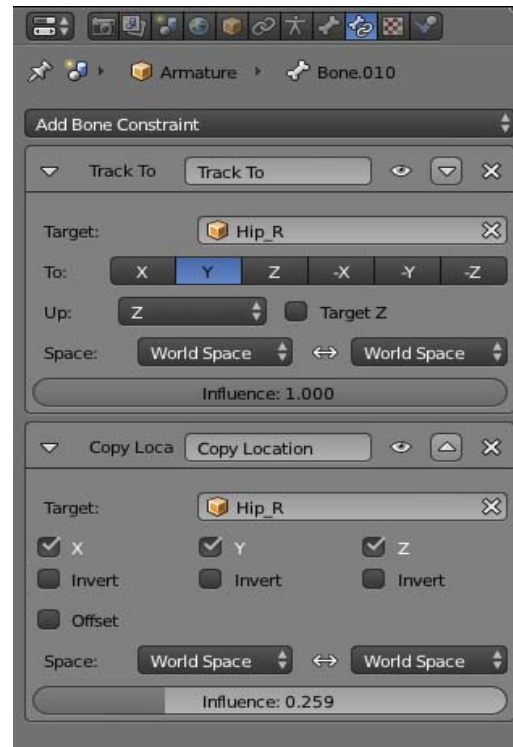
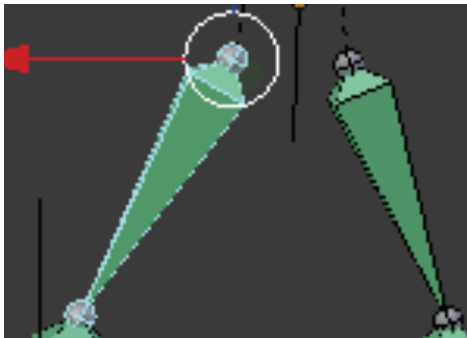


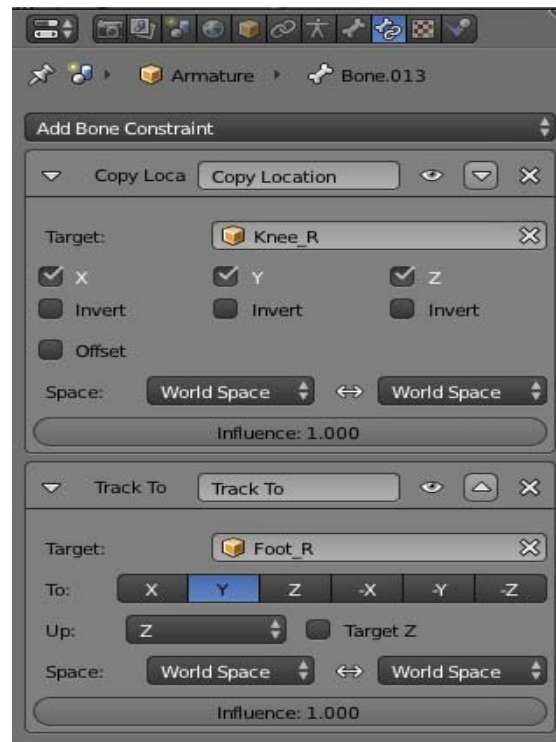
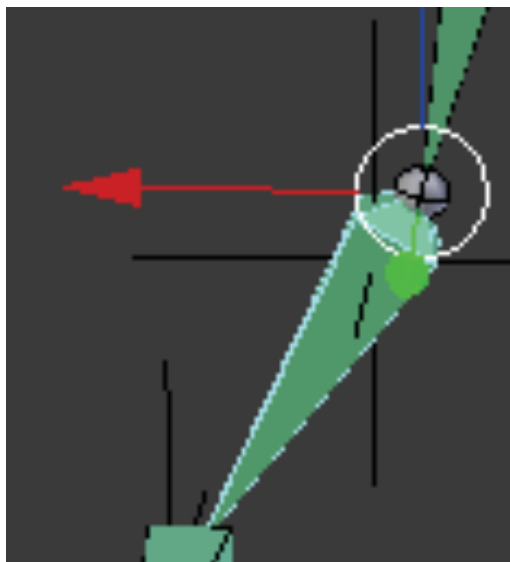
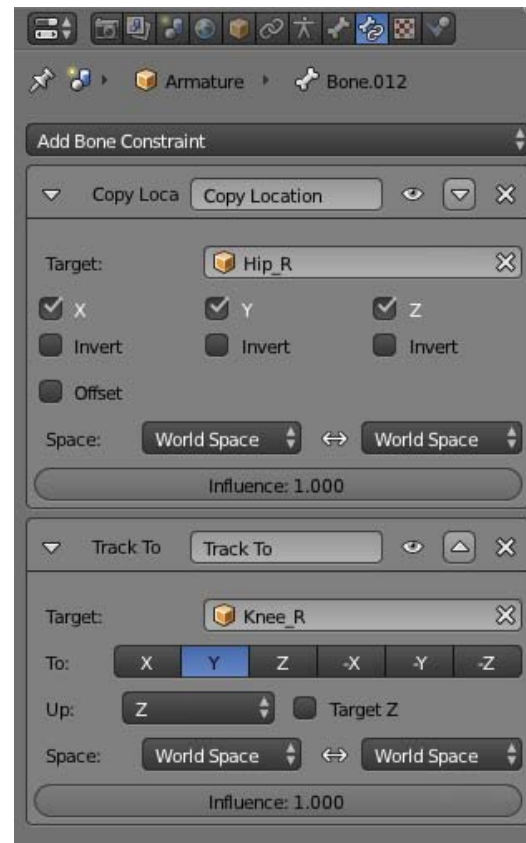
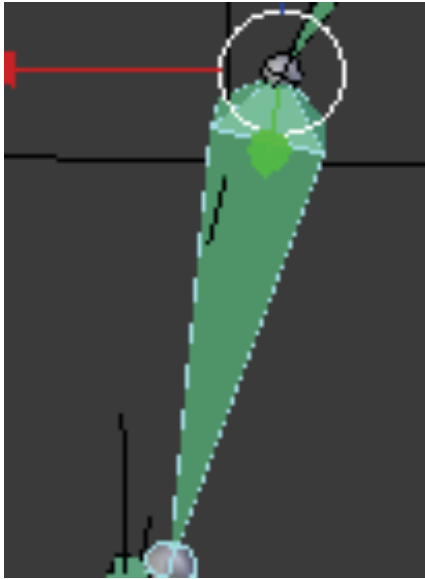
Figura T3.19. De arriba hacia abajo, el hueso Hombro, Antebrazo y Mano derechos, respectivamente y sus restricciones.

Al hombro izquierdo, el antebrazo izquierdo y la mano izquierda, se le añaden exactamente las mismas restricciones pero para los huesos del lado izquierdo, es decir, los que son “_L”.

- **Extensión de la cadera derecha, cadera derecha, rodilla derecha y pie derecho**

La llamada extensión cadera derecha, es el hueso que se creó para darle más naturalidad al movimiento del esqueleto.





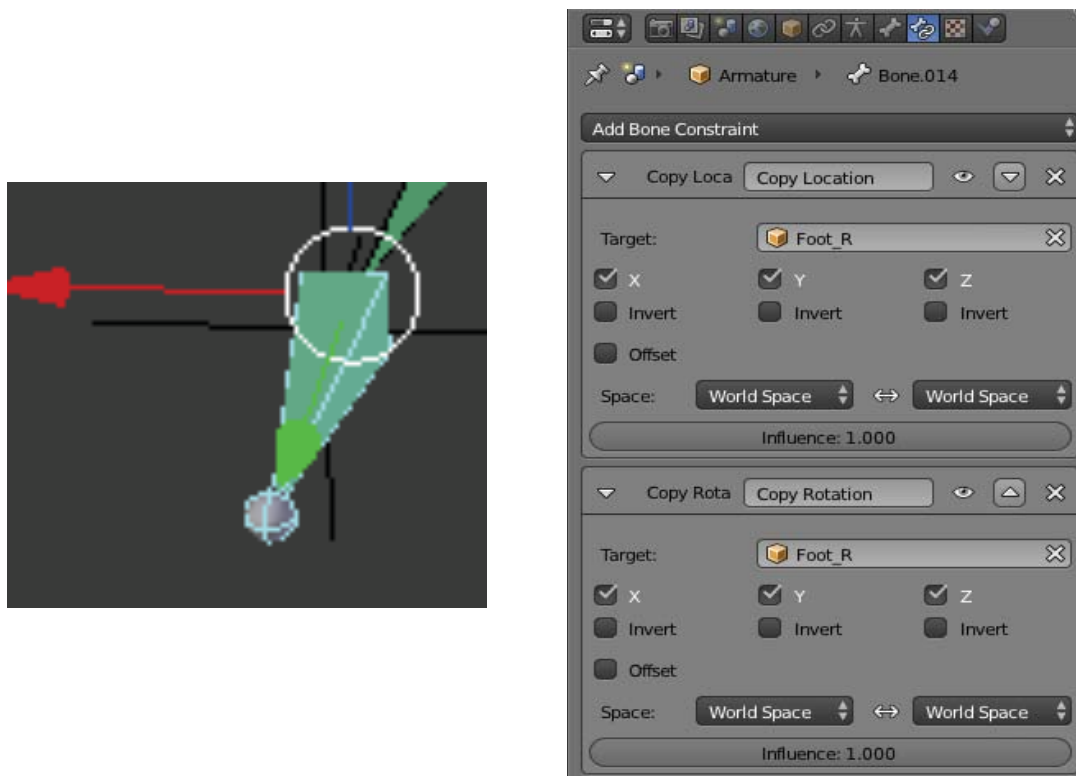


Figura T3.20. De arriba hacia abajo, Extensión de Cadera, Cadera, Rodilla y Pie derechos

Al igual que antes, se pondrán exactamente las mismas restricciones, de los huesos que forman la pierna derecha, en los huesos de la pierna izquierda, seleccionando los huesos correspondientes al lado izquierdo, los terminados en “_L”.

Lo último que queda es probar que se mueve, al igual que se hizo cuando se probó sólo con los ejes. Se guardará el proyecto y, a continuación, se ejecutará el proyecto en la carpeta Tutorial/Modelos/“guy.blend”.

3.4 Captura de movimiento de un esqueleto con una malla

Una vez abierto el proyecto “guy.blend”, Figura T3.16, se pondrá en práctica lo aprendido hasta ahora. Lo primero que se hará, será generar los ejes vacíos tal y como se hizo al comienzo de esta parte del tutorial. Una vez generados y comprobando que se mueven, se pulsará “stop” para parar de capturar el movimiento (en la columna de herramientas de Delicode Ni-Mate) y con los ejes generados se pasará a añadir las restricciones al nuevo modelo.

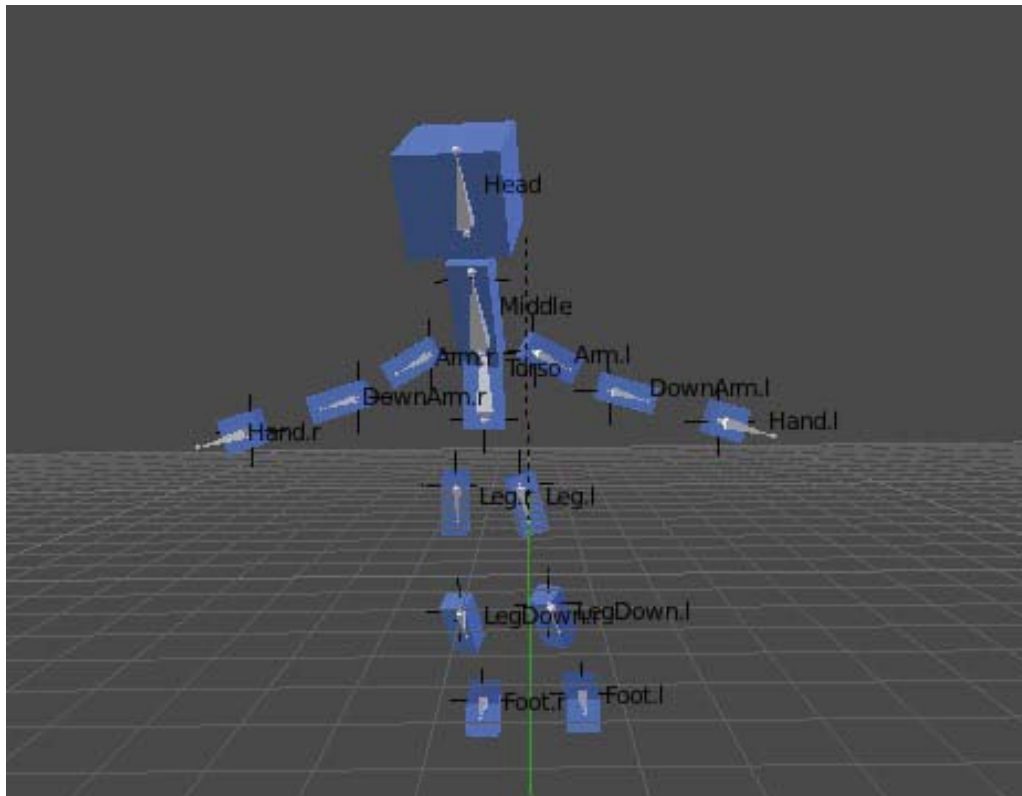


Figura T3.21. Proyecto “guy.blend”.

Se seguirán las mismas indicaciones que con el esqueleto, pero esta vez no están las extensiones de cadera. Los huesos están señalados con los nombres, con lo que será más fácil realizarla las restricciones. Si al terminar de poner las restricciones, se encuentra alguna parte que no esté bien colocada, tendrá que jugar con las restricciones de “location” y “rotation” y la influencia de estas en los huesos.

Algunas veces el Blender puede rotar los huesos, y con ello, la malla, en una dirección opuesta. Esto se soluciona con las herramientas antes mencionadas.

Por último en este tutorial, se realizará la captura de movimiento de un modelo con una malla mucho mayor. Se usará el famoso conejo creado por Blender para sus demostraciones. Es gratuito y se encuentra en la carpeta Tutorial/Modelos/Simple_Bunny.blend. Ya está preparado para moverse, simplemente se aclararán unas cuestiones.

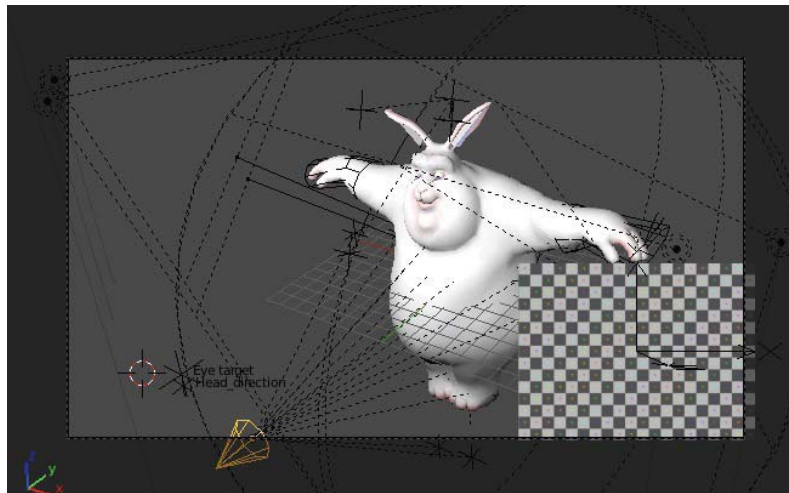


Figura T3.22. Proyecto “Simple_Bunny.blend”.

A parte, del trabajo de crear una malla de estas dimensiones, cubriendo un esqueleto. Para el movimiento, se ha realizado exactamente lo explicado en este tutorial, es decir, se han añadido los ejes vacíos, y se han creado restricciones en los huesos del modelo. También se ha de aclarar, que las restricciones antes explicadas, se pueden modificar al gusto de cada uno. Se pone el proyecto del conejo, como ejemplo, para que se vean todos estos detalles. Quitando a un lado los focos de luz, las cámaras, etc...

Tan sólo queda, pulsar “start” y probar el modelo. Dependiendo de las capacidades de cada ordenador, se verá más fluido, o menos; e incluso habrá veces que se deforme. En la carpeta Tutorial/Demo. Se encuentra un video demostrativo de cómo se mueven los modelos creados o utilizados en este tutorial.

Nota: Si se quiere crear una animación para exportarla, pulsando en el botón “rec” del timeline se podrá grabar lo que se quiera. Esto se podrá pasar a una escena ya creada, o crearla en base a este modelo, renderizar y exportarlo. Cualquier consulta que se desee hacer sobre el uso del Blender, accediendo a <http://wiki.blender.org> tendremos una gran cantidad de tutoriales con gráficas y explicaciones de los usos básicos de este software.