



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Tango:h y Kinect2.0

Tango:h and Kinect2.0

Santiago Sainz Fernández

La Laguna, 26 de Julio de 2015

Dña. **Carina Soledad González González**, con N.I.F. 54.064.251-Z profesora Titular de Universidad adscrito al Departamento de Ingeniería y de Sistemas de La Laguna, como tutor

D. **Pedro Antonio Toledo Delgado**, con N.I.F. 45.725.874-B profesor ayudante de Universidad adscrito al Departamento de Ingeniería y de Sistemas de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

“Tango:h y Kinect2.0.”

ha sido realizada bajo su dirección por D. **Santiago Sainz Fernández**, con N.I.F. 45.863.242-T.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 26 de julio de 2015.



Agradecimientos

A mi Tutora y Cotutor por mostrarse siempre atentos, comprensivos y por tener la paciencia que han tenido conmigo.

A mis amigos, en especial a Melani, Ángela y a Yeray, por apoyarme, ayudarme con diferentes aspectos del trabajo y darme ánimos en los peores momentos.

Le agradezco en especial su aportación a varias ilustraciones a Adrián Gopar. Por qué nunca sabes dónde y cómo puedes encontrar a un buen amigo.

A mis padres por su atención, su apoyo y su amor incondicional. Siempre me dais fuerzas.

En especial a mi madre porque luchando contra la enfermedad me ha enseñado lo que significa la perseverancia.

Licencia



© Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 4.0
Internacional.

Resumen

El objetivo principal del proyecto ha sido la incorporación de nuevas funcionalidades de Kinect 2.0 en Tango:h.

La aparición en el mercado del nuevo dispositivo de Kinect y la desaparición del soporte para las antiguas librerías de Tango:h, crean el marco perfecto para realizar una actualización a todo el proyecto. Gracias a él podemos expandir las posibilidades del software y estudiar la implementación de nuevas opciones de cara a reforzar Tango:h como la mejor solución en su campo.

Para llevar a cabo el trabajo se ha actualizado la versión original de Tango:h haciendo uso del software Advant. Se ha incorporado el uso de Kinect partiendo de su implementación previa en Advant y se han analizado las nuevas posibilidades del hardware para desarrollar propuestas.

El análisis se ha llevado a cabo haciendo uso de la documentación que proporciona Microsoft y examinando exhaustivamente el código de las dos herramientas de software. Además, se presentan varias propuestas que hacen uso de las nuevas capacidades del dispositivo para su implementación en Tango:h en un futuro. Entre ellas se incluye información sobre el aporte en materia de mejoras gestuales y faciales.

Palabras clave: Tango:h, Advant+, Kinect 2.0, Rehabilitación, Reconocimiento facial, Reconocimiento emocional, Juegos para la Salud.

Abstract

The main objective of the project has been the incorporation of new functionalities that Kinect 2.0 has and add them to Tango:h.

The appearance on the market of the new Kinect device and the disappearance of the support for older libraries that was using Tango:h, are the setting for an upgrade to the entire project. Thanks to it we can expand the possibilities of the software and to study the implementation of new options in order to strengthen Tango:h as the best solution in its field.

To accomplish the work we has updated the original version of Tango:h using Advant. We has incorporated the use of Kinect based on their previous deployment in Advant and analyzed the new possibilities of the hardware to in order to develop proposals.

The analysis was conducted using the documentation provided by Microsoft and thoroughly examining of the both software tools. In addition, several proposals that make use of the new capabilities of the device for deployment in Tango in the future are presented. Including information about the contribution them are included on gestures and facial enhancements.

Keywords: *Tango:h, Advant+, Kinect 2.0, Rehabilitation, Face Recognition, Emotional Recognition, Games for Health.*

Índice General

Capítulo 1. Introducción	5
1.1 Contexto del proyecto.....	5
1.2 Objetivos del proyecto	6
Capítulo 2. Antecedentes y estado inicial del proyecto	8
1.1 Tango:h	8
1.2 Advant	11
1.3 Tecnología	11
1.3.1 Hardware	11
1.3.2 Software.....	12
Capítulo 3. Análisis, Diseño e Implementación de Tango:h 2.0	15
3.1 OpenNI – Kinect SDK.....	15
3.2 Actualizaciones de Advant	17
Capítulo 4. Propuestas para la mejora de la interacción gestual	24
4.1 Detección de la mano.....	24
4.1.1 Detección de la mano.....	24
4.1.2 El estado Lasso.....	26
4.2 Diseño de propuestas	27
4.2.1 Detección de la mano.....	27
4.2.2 El estado Lasso.....	35
4.3 Prototipo de detección de mano	38
4.3.1 Pruebas de caso de estudio	38
4.3.2 Pruebas de profundidad	39

Capítulo 5. Incorporación de información facial y emocional	39
5.1 Propuestas	45
Capítulo 6. Conclusiones y líneas futuras	49
Capítulo 7. Summary and Conclusions	51
Capítulo 8. Presupuesto	53
Bibliografía	54

Índice de figuras

Figura 1. Logotipo Tango:H	8
Figura 2. Logotipo ADVANT.....	11
Figura 3. Dispositivo Kinect 2.0.	12
Figura 4. Logotipo OpenNI.....	13
Figura 5. Logo Kinect for Windows SDK	13
Figura 6. Pseudocódigo de Medición de Distancias	19
Figura 7. Pseudocódigo comprobar distancia al sensor.....	19
Figura 8. Diagrama de comprobación de distancia.....	20
Figura 9. Panel Rango de Acción 100	21
Figura 10. Panel Rango de Acción 150	21
Figura 11. Ejemplo de Rango de los Objetivos.....	22
Figura 12. Panel de tiempo.....	23
Figura 13. Puntos identificativos de Kinect	25
Figura 14. Posición de mano en estado lasso	26
Figura 15. Diagrama de ejercicio de posicionamiento de objetos.....	28
Figura 16. Ejemplo de posicionamiento de objetos	29
Figura 17. Diagrama de ejercicio de secuencias	32
Figura 18. Ejemplo de secuencias de manos.....	33
Figura 19. Ejemplo de unir objetos	37
Figura 20. Profundidad de objeto por la derecha.....	39
Figura 21. Profundidad de objeto por la izquierda	40
Figura 22. Prueba profundidad Kinect 1.....	40
Figura 23. Prueba profundidad Kinect 2.....	41
Figura 24. Reconocimiento facial con Kinect	43
Figura 25. Programa SHORE en funcionamiento.....	46
Figura 26. Reconocimiento emocional con CrowdEmotion.....	47

Índice de tablas

Tabla 1. Articulaciones en las librerías	16
Tabla 2. Estados de la mano	25
Tabla 3. Propiedades de la cara para Kinect	43
Tabla 4. Valores de las propiedades de la cara	44
Tabla 5. Ejercicios rehabilitación facial.....	48
Tabla 6. Tabla de presupuesto	53

Capítulo 1.

Introducción

1.1 Contexto del proyecto

Son muchos los estudios académicos al respecto que demuestran el potencial que poseen los videojuegos tanto para la ayuda al aprendizaje como para la socialización o la promoción de una vida saludable. Una investigación realizada por Casey (1992) destaca las cualidades positivas de los videojuegos en cuanto al desarrollo de destrezas y capacidades de tipo multisensorial, auditivas, visuales y kinestésicas.

A ello se suma la utilización de diferentes tipos de tratamientos para la rehabilitación de adultos usando videojuegos. En 2011 se hace uso por primera vez en Argentina de técnicas de rehabilitación en el área motor y cognitiva de los pacientes, en este caso de la tercera edad. También se había comenzado a implementar con éxito en los centros hospitalarios norteamericanos, como el Herrin o el Insituto de Rehabilitación de Glenrose. Estos tipos de tratamientos son reconocidos y fomentados por la Organización Mundial de la Salud, en su Guía Regional para la Promoción de la Actividad Física, enunciando en él que los videojuegos son una buena herramienta, en tanto proporcionen el movimiento necesario para considerarse una actividad física como cualquier otra.

En el caso concreto de Tango:H se trata de una herramienta valorada por entidades como el Colegio San Fernando Duggi. De Santa Cruz de Tenerife, el Aula Hospitalaria del Hospital Universitario de Canarias (HUC) o el Aula Hospitalaria del Hospital Dr. José Molina Orosa de Lanzarote. Además la herramienta está siendo utilizada en la asociación Down Tenerife (Santa Cruz de Tenerife).

Para conseguir la mayor versatilidad y accesibilidad posible al software se decidió dividirlo en dos partes bien diferenciadas. La primera sería la orientada directamente a los niños (Tango:h) y la segunda concebida para los especialistas (Tango:h Designer) como médicos.

Kinect es un dispositivo de hardware que es capaz de rastrear el “esqueleto” de una persona y seguir sus movimientos. Gracias a ello se pueden trasladar dichos movimientos a un software que permita la interacción entre un usuario y lo que se le muestra a través de un monitor. Esta es la tecnología de la que parte Tango:h para llevar a cabo su cometido como herramienta de rehabilitación para pacientes.

Pero han pasado varios años desde la presentación del proyecto inicial y la tecnología ha avanzado, el dispositivo de hardware Kinect ha quedado obsoleto frente a su nueva versión Kinect 2.0. Las diversas mejoras en rendimiento, fiabilidad y funcionalidades hacen que el cambio de la primera versión a esta segunda sea el paso lógico a realizar.

1.2 Objetivos del proyecto

El **objetivo principal** del proyecto es el de la incorporación de nuevas funcionalidades de Kinect 2 a Tango:h.

Los **objetivos específicos** del proyecto han sido:

- Incorporación de soporte del SDK Kinect 2 al proyecto Tango:h a partir del software Advant.
- Análisis del nuevo sistema y diseño de propuestas.
- Mejora de interacción gestual
- Estudio de incorporación facial y emocional.

La memoria ha sido estructurada con un capítulo 2 centrado en los antecedentes y estado actual del proyecto; un capítulo 3 donde se explican las fases de análisis, diseño e implementación correspondiente al primer objetivo

específico; un capítulo 4 que trata sobre las diferentes mejoras en el ámbito gestual con diferentes propuestas; y un capítulo 5 que aborda las mejoras en materia facial que aporta el nuevo dispositivo.

Capítulo 2.

Antecedentes y estado inicial del proyecto.

2.1 Tango:h



Figura 1. Logotipo Tango:H

Tango:H (Tangible Goals: Health) es una plataforma creada en 2012 y destinada a niños hospitalizados con diversidad funcional desarrollada mediante un convenio de colaboración entre el Instituto Tecnológico y de Energías Renovables (ITER) y el Grupo de Investigación, Interacción, Tecnología y Educación (i-TED) del Departamento de Ingeniería de Sistemas, Automática y Arquitectura y Tecnología de Computadores de la Universidad de La Laguna.

Mediante la utilización del Kinect de Microsoft, Tango:h permite la interacción con sistemas de información sin necesidad de mantener contacto físico con los sistemas de control tradicionales.

A continuación explicaremos cómo está estructurado Tango:h y a definir diferentes conceptos necesarios para la comprensión de algunos de los aspectos tratados en el proyecto y la memoria. Siempre basado en la documentación de la herramienta accesible desde su página web y desde la bibliografía (Manual de Tango:H Designer y Manual de Usuario).

La plataforma Tango:h dispone, además del modo tradicional de un jugador, un modo multijugador (secuencial, competitivo y colaborativo), capaz de detectar dos cuerpos humanos de manera simultánea. El administrador integrado permite gestionar a los usuarios, agruparlos y asignar ejercicios. Una vez realizados los ejercicios, estos quedarán almacenados, de forma que se podrán recuperar análisis estadísticos sencillos y comprobar en cualquier momento cómo se desarrollaron. Todo bajo una interfaz sencilla e intuitiva que ha sido orientada hacia los más pequeños.

El potencial de Tango:h reside en su capacidad de generar ejercicios, es decir, no es una plataforma estática en la que los ejercicios o juegos están completamente definidos e integrados, sino que permite la implementación de estos mediante un configurador que hace sencilla esta tarea. Esta característica ofrece la posibilidad de crear ejercicios adaptados a las necesidades de los usuarios y servir como herramienta para la mejora de la calidad de vida de personas con diversidad funcional. Este configurador de ejercicios se denomina Tango:h Designer.

Los ejercicios en Tango:H se dividen en tres categorías principales. Físicos, Cognitivos y Libres:

- **Físicos:** Pensados principalmente para la rehabilitación de personas con movilidad reducida o problemas físicos. El objetivo es alcanzar con las manos los diferentes objetivos que van apareciendo en la pantalla.
- **Cognitivos:** En estos ejercicios el jugador utilizará sus manos pero en vez de hacer uso de las ayudas por colores de los ejercicios físicos tendrá que hacer uso de “pistas” (en forma de sonidos, etc...) que le ayuden a alcanzar los objetivos reforzando los procesos mentales como la memoria, el razonamiento lógico o la percepción. Dentro de este tipo de ejercicios hay tres subtipos más, *emparejamiento* (mejora la memoria y percepción); *ordenación* (mejora el aprendizaje y el razonamiento lógico); y *clasificación* (mejora el razonamiento lógico).
- **Libres:** Como su nombre indica son ejercicios que no poseen restricciones. Su ventaja consiste en ser capaces de mezclar objetivos

de tipo físico y cognitivo, sirviendo para ejercitarse en un plano más global durante el ejercicio.

Las partes de un ejercicio comprenden los siguientes aspectos cuyas definiciones es conveniente tener en cuenta:

- **Objetivo:** Se trata de un elemento que el usuario debe alcanzar con uno o más “puntos de contacto”. Un objetivo puede estar compuesto por una imagen o una región determinada de la pantalla.
- **Fase:** Agrupación de objetivos. Para superarla el jugador debe alcanzar todos los objetivos.
- **Paso:** Agrupación de fases. Para superar un paso el jugador debe haber completado todas las fases.
- **Ejercicio:** Un ejercicio consta de un conjunto de Pasos.

Tango:H posee varios modos de juego orientados al número de jugadores que van a hacer uso del programa:

- **Individual:** El jugador realizará los ejercicios en solitario.
- **Multijugador:** Es el modo de juego que permite realizar los ejercicios a dos jugadores. Dentro de este modo hay varios subtipos:
 - *Secuencial:* Los jugadores se enfrentan al mismo ejercicio uno a continuación del otro.
 - *Colaborativo:* Con el objetivo de ayudarse los jugadores se enfrentan al mismo ejercicio de manera simultánea.
 - *Competitivo:* Los dos jugadores realizan el mismo ejercicio de forma simultánea enfrentándose por ver quién de los dos termina antes el ejercicio.

2.2 Advant



Figura 2. Logotipo ADVANT

A la par que el proyecto Tango:H el ITER ha desarrollado en colaboración con SINPROMI un software denominado ADVANT (Advanced Therapeutics), a nivel de programación las dos plataformas tienen muchas similitudes y por eso ha sido escogida para desarrollar parte del trabajo. Ambas “plataformas” hacen uso del mismo dispositivo (Kinect) e incluso están divididas de la misma manera, una parte del programa ejecutando ejercicios y la otra que sirve para diseñar los susodichos. En este caso el configurador de ejercicios se denomina ADVANT-Editor.

2.3 Tecnología

2.3.1 Hardware

Kinect es un nuevo tipo de controlador para juegos y entretenimiento creado por Alex Kipman y desarrollado por Microsoft, pensado en primer lugar para la consola Xbox 360 y desde junio de 2011 para PC a través de Windows 7 y Windows 8.

Kinect permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional (un mando), mediante una interfaz natural de usuario que reconoce gestos, comandos de voz, y objetos e imágenes. El sensor de Kinect está diseñado para ser colocado por encima o por debajo de una pantalla de televisión o un monitor.



Figura 3. Dispositivo Kinect 2.0

El dispositivo (Kinect 1) cuenta con una cámara RGB, un sensor de profundidad, un micrófono de múltiples matrices y un procesador personalizado que ejecuta el software patentado, que proporciona captura de movimiento de todo el cuerpo en 3D, reconocimiento facial y capacidades de reconocimiento de voz. El 21 de Mayo de 2013 Microsoft anunció su Kinect 2.0 que incluye diversas mejoras respecto a la anterior versión.

2.3.2 Software

Ambas plataformas están desarrolladas haciendo uso de la misma tecnología. Usando .NET como framework y C# como lenguaje de programación. La base de datos se ha creado haciendo uso de la tecnología SQL mientras que XML ha sido la escogida para la parte gráfica y de almacenamiento de información en ficheros.

El IDE (Integrated Development Enviroment) para su desarrollo original ha sido Microsoft© Visual Studio 2010 aunque para llevar a cabo el proyecto se ha utilizado el mismo IDE en su versión 2013.

Originalmente Tango:h y Advant hacían uso de OpenNI SDK (Open Natural Interaction). OpenNI era una organización sin ánimo de lucro que promovía la interacción entre usuarios y diferentes dispositivos. Para ese fin desarrollaron el framework OpenNI SDK. Dicho framework proporcionaba

varias APIs de código abierto que permitían la comunicación del software Tango:h con el dispositivo de hardware Kinect.



Figura 4. Logotipo OpenNI

El proyecto se discontinuó quedando únicamente un repositorio sin actualizar cuando empezó a gestarse este proyecto. Debido a esto el SDK no ofrece soporte para el nuevo hardware (Kinect 2.0) ni tampoco es compatible con las últimas versiones de .NET. Esto refuerza la decisión de cambiar de un SDK obsoleto y sin soporte, a un SDK libre y que está directamente vinculado a Microsoft, responsable directo del hardware.



Figura 5. Logo Kinect for Windows SDK

El SDK oficial proporciona las herramientas y APIs necesarias para desarrollar aplicaciones que hagan uso de Kinect en un entorno para Microsoft Windows y completamente compatible con las últimas versiones de .NET.

A parte de estas tecnologías de software ya implementadas en el proyecto, también se busca un sistema recomendador que sea capaz de detectar las emociones de los usuarios. Los sistemas recomendadores o de recomendación

usan una técnica específica de filtro de información para presentar distintos tipos de información que pueden interesar a un usuario en particular. En estos términos haría referencia a un sistema capaz de recomendar un ejercicio u otro en función no solo de las habilidades del usuario si no de su estado de ánimo.

Capítulo 3.

Análisis, Diseño e Implementación de Tango:h 2.0.

El proyecto completo se dividía en Tango:h (el módulo de ejercicios) y Tango:h Designer. Se decidió comenzar por los cambios del módulo de ejercicios pues era la parte del proyecto que más importancia tenía de cara a añadir el nuevo hardware.

3.1 OpenNI – Kinect SDK.

La fase de análisis y diseño fue bastante atípica. El ITER proporcionó su software Advant con las últimas mejoras realizadas, entre ellas la incorporación del nuevo SDK. Por tanto el trabajo llegado a ese punto fue el de analizar las diferencias entre el código de Tango:h y Advant.

Ambas herramientas son bastante similares por lo que usando un comparador de código se fueron realizando los cambios en los diferentes archivos. La fase de análisis y diseño fue prácticamente nula y se abordó casi directamente la parte de implementación.

Los principales cambios entre el SDK de Microsoft y el de OpenNI fueron las referencias a librerías y el cambio en la referencia de las articulaciones. La siguiente tabla compara la manera de referenciar las articulaciones para cada uno de las librerías.

OpenNI	Kinect 2 SDK
SkeletonJoint.Head	JointType.Head
SkeletonJoint.LeftShoulder	JointType.ShoulderLeft
SkeletonJoint.RightShoulder	JointType.ShoulderRight
SkeletonJoint.LeftElbow	JointType.ElbowLeft
SkeletonJoint.RightElbow	JointType.ElbowRight
SkeletonJoint.LeftHip	JointType.HipLeft
SkeletonJoint.RightHip	JointType.HipRight
SkeletonJoint.LeftKnee	JointType.KneeLeft
SkeletonJoint.RightKnee	JointType.KneeRight
SkeletonJoint.LeftHand	JointType.HandLeft
SkeletonJoint.RightHand	JointType.HandRight
SkeletonJoint.LeftFoot	JointType.FootLeft
SkeletonJoint.RightFoot	JointType.FootRight
SkeletonJoint.Torso	JointType.SpineMid

Tabla 1. Articulaciones en las librerías.

A parte de estos cambios también se implementaron dos nuevos archivos para añadir funciones que antiguamente ya gestionaba la librería de OpenNI. Ahora es necesario manejar ciertos aspectos de manera manual con el nuevo SDK.

El primer fichero importante es el ScaledSkeleton.cs cuyo objetivo es el de calcular las coordenadas de las articulaciones y devolverlas en forma de punto.

Por último también se creó el fichero Point3D.cs que define la estructura de un punto en 3D con coordenadas X, Y, Z. Posee además varias funciones útiles para crear los puntos de diferentes maneras como inicializándolo a 0 o para copiar otro existente con las mismas coordenadas. Este Point3D ya se usaba en la última versión de Tango:h 1 pero estaba definido y gestionado por la librería, lo mismo ocurre con el fichero para el correcto seguimiento del esqueleto.

3.2 Actualizaciones de Advant.

Una de las tareas a realizar fue la de incorporar las nuevas funciones que tuviese Advant a Tango:h 2.0. Primero se identificaron dichas funciones, Medición de la distancia al sensor, Rango de Acción (actionRange) y el tiempo (time). La primera de las funciones está ubicada en la parte de Tango:h encargada de ejecutar los ejercicios, mientras que las dos restantes son funciones añadidas al Designer.

Medición de distancia al sensor (Análisis):

En la fase de análisis se estudió el alcance de la nueva función y su objetivo.

El dispositivo Kinect es una herramienta versátil pero como cualquier sistema de detección de movimientos existe una distancia óptima para que la cámara detecte al usuario. Es debido a esto que se ha implementado la función que permite a Advant y ahora también a Tango:h 2.0, informar al usuario si se encuentra o no a la distancia óptima y si debe alejarse o acercarse para posicionarse de forma correcta.

El SDK de Kinect no posee una función que mida directamente la distancia que existe entre el jugador y el hardware por tanto la función se ha

implementado de manera manual. Las funciones necesarias se encuentran en un fichero nuevo, creado para añadir esta funcionalidad al proyecto.

La nueva clase se encarga de calcular la altura del esqueleto del jugador que está frente al sensor. Para ello se tienen en cuenta tres datos importantes:

- **El margen:** Como su propio nombre indica, representa al margen que se usa para calcular el límite de la zona óptima. Si no existiesen estos márgenes el jugador tendría que colocarse a la distancia exacta sin errores, lo cual resultaría incómodo y poco práctico.
- **La altura del esqueleto:** Es la altura media reflejada en metros de un esqueleto. Para una altura de pies a cabeza el valor sería de 1.80 o 0.80 si calculásemos de tronco a cabeza (en caso de que, por ejemplo, el jugador estuviese sentado).
- **Distancia al sensor:** La distancia óptima a la que debe estar el jugador con respecto al sensor (en metros).

Para el proyecto se ha escogido un 0.25 (metros) de **margen**. Para los otros dos datos se ha tenido en cuenta la recomendación de Microsoft. Según su documentación para una persona (de pie) que mida 1.80 metros la distancia óptima al sensor será de 2 metros.

Teniendo en cuenta estos datos la nueva clase detecta si la persona está sentada o de pie en función de la articulación desde la que puede empezar a medir. Una vez detectado cualquiera de los dos pies, se procede a calcular la distancia que existe entre esa articulación y la siguiente hasta llegar a la cabeza. Cuando se ha terminado de medir la distancia a la última articulación la suma total resultante será la altura del esqueleto del jugador.


```

DistanciaEntre2Articulaciones (Articulacion1, Articulacion 2) {
    Devuelve RaizCuadrada(
        ((Articulacion1.X - Articulacion2.X)^2) +
        ((Articulacion1.Y - Articulacion2.Y)^2) +
        ((Articulacion1.Z - Articulacion2.Z)^2))
}

DistanciaEntreTodasArticulaciones (ArrayArticulaciones) {
    LongitudTotal = 0
    Desde Indice = 0 mientras Indice < (TamañoArrayArticulaciones -1) e Indice++ Hacer
        LongitudTotal += DistanciaEntre2Articulaciones(ArrayArticulaciones[Indice],
            ArrayArticulaciones[Indice+1])

    FinHacer
    Devuelve LongitudTotal
}

```

Figura 6. Pseudocódigo de Medición de Distancias.

Una vez obtenida la altura del esqueleto, se calcula el límite inferior, el límite superior y la zona óptima a la que debe encontrarse el usuario a través de una sencilla regla de tres como muestra el pseudocódigo justo debajo.

```

ComprobarDistancia (Altura, Distancia) {
    ZonaOptima = (Altura * DistanciaMedia) / AlturaMedia;
    LimiteInferior = ZonaOptima - Margen;
    LimiteSuperior = ZonaOptima + Margen;

    Si (Distancia > LimiteSuperior) entonces Aproximarse al Sensor FinSi
    SiNo Distancia < LimiteInferior) entonces Alejarse del Sensor FinSi
    SiNo Jugador dentro de la Zona óptima FinSi
}

```

Figura 7. Pseudocódigo comprobar distancia al sensor

El funcionamiento general de esta nueva característica del software puede resumirse en el siguiente diagrama.

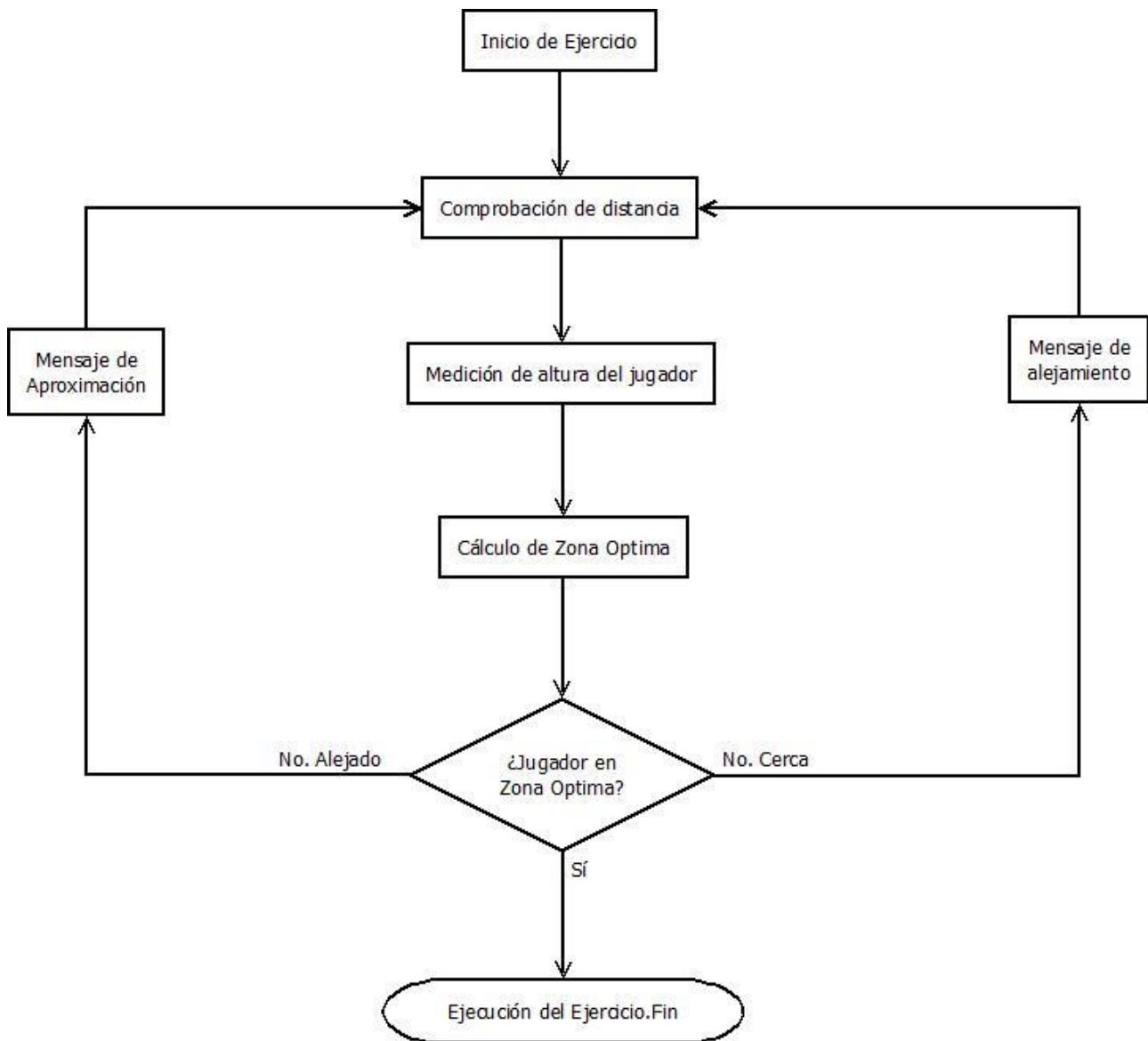


Figura 8. Diagrama de comprobación de distancia.

Medición de distancia al sensor (Diseño e Implementación):

La fase de diseño para esta función es nula. Debido a la similitud entre ambos códigos no hizo falta diseñar el proceso de implementación.

Se añadieron los archivos necesarios al proyecto, el archivo de *Altura.cs* que contiene todas las funciones necesarias para el cálculo de la altura tal y como se explica en el apartado de análisis. Se añadieron los archivos necesarios para el diseño de la nueva pantalla de aviso (mensajes de acercamiento y alejamiento), estos son *ScreeInfo.cs* y *ScreenInfo.Desginer.cs*. Por último se añadió la tecla “Insert” para iniciar el proceso de cálculo de la altura del usuario, dicha asignación se realiza en el fichero *MainWindows.cs*

Rango de Acción (Análisis):

En la fase de análisis se estudió el alcance de la nueva función y su objetivo.

En el editor Tango:h Designer, se ha añadido la posibilidad de especificar el rango de los objetivos. En la versión inicial del proyecto el rango de acción, en términos de porcentaje, era del 100%. Es decir, el tamaño del objetivo era el rango de acción del mismo y este no podía modificarse.

Con esta nueva característica el mínimo (valor por defecto) queda establecido como 100 y puede aumentarse hasta lo que el diseñador del ejercicio desee. De tal forma que un objetivo sea más fácil de alcanzar o más difícil. Esto abre la puerta a ejercicio ligeramente más complejos y por lo tanto más ajustados a las necesidades del paciente.

Hay que tener en cuenta que la variable del rango de acción de los objetivos es global. Si el rango definido fuese de 150%, lo será para todos los objetivos al mismo tiempo.

Rango de Acción (Diseño e Implementación):

La fase de diseño para esta función es nula. Debido a la similitud entre ambos códigos no hizo falta diseñar el proceso de implementación.

Se añadieron las líneas de código necesarias en el Editor, modificando el archivo de estructura de datos de los ejercicios (Exercise.cs) para añadir la nueva variable. También se modificó el VisualEditorForm.cs para implementar en la interfaz la nueva opción. El resultado puede verse en las siguientes imágenes.

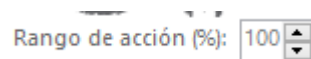


Figura 9. Panel Rango de Acción 100

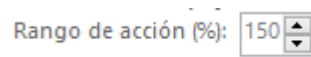


Figura 10. Panel Rango de Acción 150

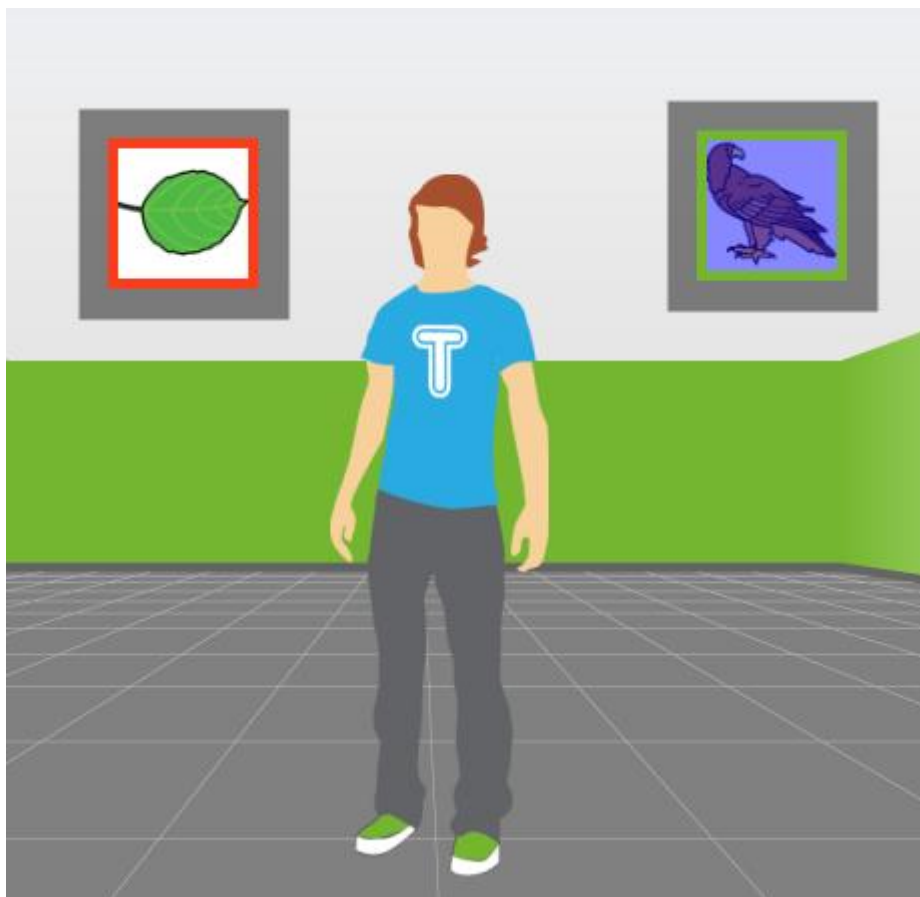


Figura 11. Ejemplo de Rango de los Objetivos.

El rango de los objetivos viene expresado por el cuadrado oscuro que rodea a los objetivos (imagen del pájaro y la hoja).

También se llevaron a cabo modificaciones en el .dtd y en el XMLParser para incluir la información de la nueva variable en los archivos de ejercicios que luego han de exportarse al otro módulo de Tango.

Tiempo (Análisis):

En la fase de análisis se estudió el alcance de la nueva función y su objetivo.

En el editor Tango:h Designer, se ha añadido la posibilidad de especificar el tiempo de los objetivos. Ahora puede establecerse el tiempo en segundos para los objetivos. Esto permite una mejor adaptación del ejercicio a un tipo de paciente o tipo de rehabilitación concreta de tal forma que puede darse más o menos tiempo de reacción para el jugador.

El tiempo tiene un valor mínimo y por defecto de 1,0 y hace referencia a los segundos que se quieran especificar.

Tiempo (Diseño e Implementación):

La fase de diseño para esta función es nula. Debido a la similitud entre ambos códigos no hizo falta diseñar el proceso de implementación.

Se añadieron las líneas de código necesarias en el Editor, modificando el archivo de estructura de datos de los ejercicios (Exercise.cs) para añadir la nueva variable. También se modificó el VisualEditorForm.cs para implementar en la interfaz la nueva opción. El resultado puede verse en las siguientes imágenes.

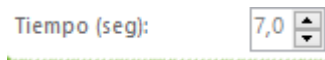


Figura 12. Panel de tiempo.

También se llevaron a cabo modificaciones en el .dtd y en el XMLParser para incluir la información de la nueva variable en los archivos de ejercicios que luego han de exportarse al otro módulo de Tango.

Capítulo 4.

Propuestas para la mejora de la interacción gestual.

Uno de los requisitos del trabajo realizado es la investigación de las nuevas posibilidades que ofrece el dispositivo y como pueden mejorar la herramienta y la calidad del programa de cara a los usuarios.

A continuación se exponen varias mejoras que podrían aplicarse a Tango:h en líneas futuras.

4.1 Análisis de mejoras.

4.1.1 Detección de la mano.

En la versión anterior del dispositivo (el primer Kinect) las manos estaban representadas para el software del tracker como dos pelotas (bolas), una por cada mano. Es decir, que no se podía tener información detallada de los movimientos concretos de la mano.

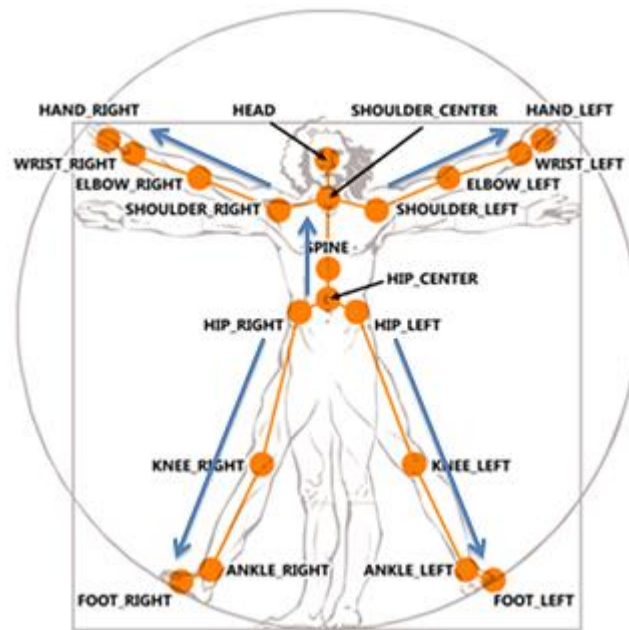


Figura 13. Puntos identificativos de Kinect.

El nuevo dispositivo es ahora capaz de identificar varios estados de la mano en función principalmente del pulgar de cada una de ellas (MSDN Microsoft). Estos nuevos estados son:

Estados	Descripción
Desconocido (Unknow)	Se desconoce el estado de la mano.
Sin Rastrear (NotTracked)	No se está rastreando (siguiendo) la mano.
Abierta (Open)	El usuario tiene la mano abierta.
Cerrada (Closed)	El usuario tiene la mano cerrada.
Lasso	Este estado de la mano es el más complejo y será descrito en el siguiente sub-apartado de este capítulo.

Tabla 2. Estados de la mano.

4.1.2 El estado Lasso.

Como se indica en el sub-apartado anterior, lasso es uno de los estados que Kinect 2.0 es capaz de asignarle a la mano. Es de lejos el estado más complejo de todos los posibles.

En principio el estado lasso hace referencia a una mano con los dedos índice y corazón estirados. Como una mano haciendo el símbolo de la paz pero con el corazón y el índice juntos en vez de separados. Aunque si la mano es lo suficientemente larga y está lo suficientemente cerca del sensor el seguimiento puede llegar a funcionar solo con el dedo índice ya que el segundo dedo (el corazón) se usa para crear algo lo suficientemente grande para ser reconocido por el sensor.



Figura 14. Posición de mano en estado lasso.

No existen en la documentación de Microsoft ninguna regla de uso para el estado lasso. Pero señalan que puede ser usado en mecánicas gestuales como:

- Seleccionar un área usando un movimiento circular.
- Activar una región, como un objeto en pantalla.
- Dibujar en la pantalla.
- Girar un objeto en pantalla.

4.2 Diseño de propuestas.

4.2.1 Detección de la mano.

Si nos quedásemos exclusivamente con los estados más sencillos, mano abierta y mano cerrada, se podrían aplicar para realizar nuevos planteamientos en los ejercicios y nuevos ejercicios en sí mismos. Las propuestas son las siguientes:

- **Ejercicio de posicionamiento de objetos:** En este tipo de ejercicios el jugador deberá “recoger” una serie de objetos en un límite de tiempo y dejarlos en un área objetivo específica de la pantalla. Para coger los objetos la mano debe posicionarse sobre ellos abierta y deberá cerrarse para “cogerlo” y arrastrarlo. Las metas del ejercicio pueden variar, pueden ser absolutas, es decir, todos los objetos deben quedar dentro del área específica en el tiempo designado. O pueden ser parciales, pueden establecerse un número mínimo de objetos a depositar y usar el resto como puntuación extra.

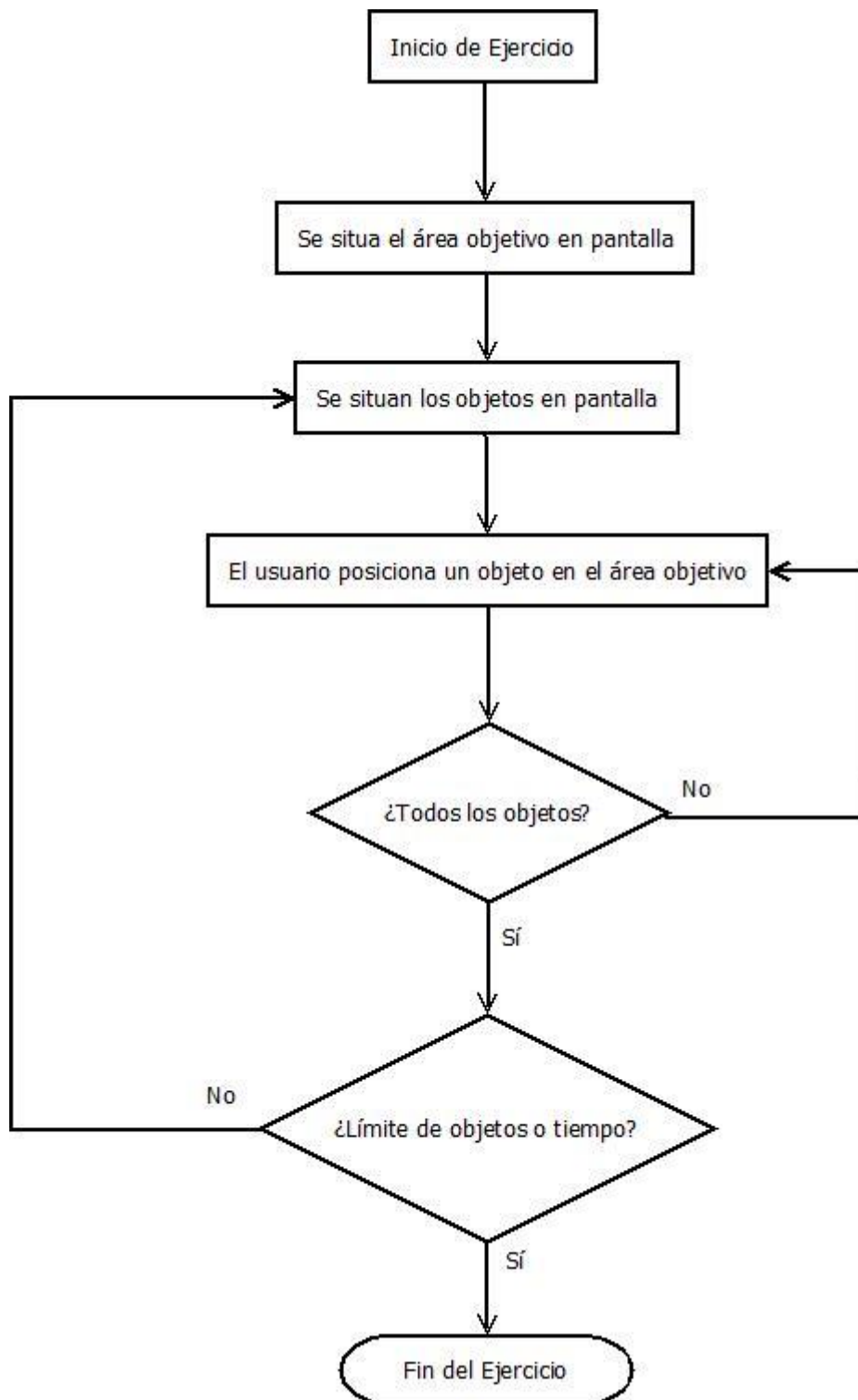


Figura 15. Diagrama de ejercicio de posicionamiento de objetos.

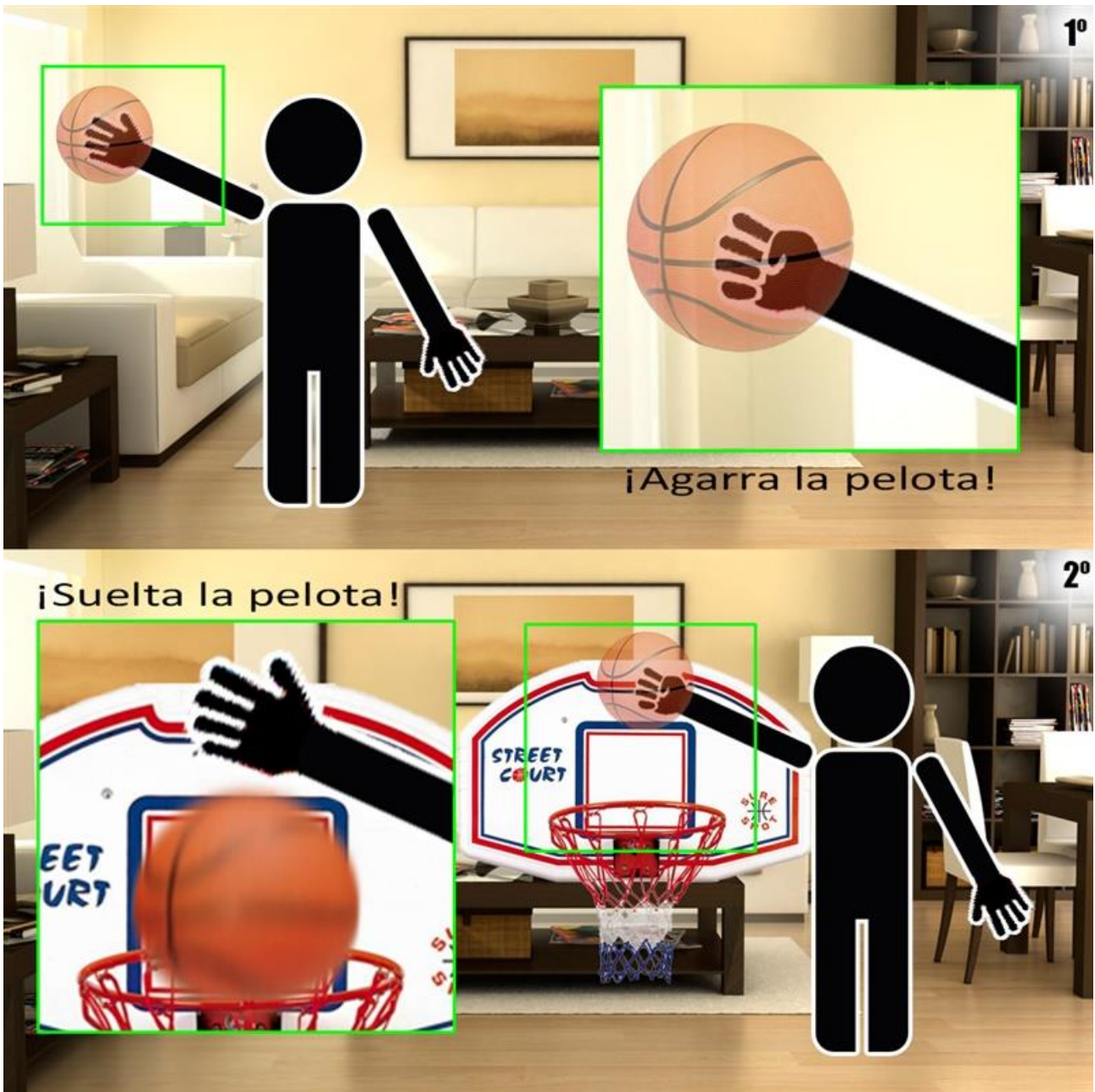


Figura 16. Ejemplo de posicionamiento de objetos.

Diseño de la implementación:

Lo primero que debemos tener en cuenta es que los ejercicios en Tango:h están basados en objetivos estáticos que no poseen movimiento. Por eso lo principal es crear un nuevo tipo de objetivo que posea dos estados, fijo y en movimiento. Cuando definamos el objetivo se nos dará la opción de escoger de qué tipo queremos que sea y entre esas opciones deberemos escoger el nuevo tipo al que llamaremos “Movable” y las articulaciones válidas para interactuar con él. Dichas articulaciones deben estar limitadas a las manos, puede ser una

en concreta o las dos y se seleccionarían como normalmente se hace en el editor.

El objetivo “Movable” tendría dos estados, uno fijo en el que su estado sería como el de cualquier objetivo normal (es decir, quieto) y otro en movimiento. El modo en movimiento se activaría al pasar la mano correcta sobre el objetivo y el estado de la articulación pasase de “Abierta” a “Cerrada”, mientras el estado de la mano sea el de “Cerrada” el objetivo actualizaría su ubicación con las mismas coordenadas que sirven para ubicar la mano (haciendo uso del punto 3D/2D de posición de la articulación), de esta forma el objetivo parecería pegado a la mano del usuario. Si la mano estuviese cerrada no debe activarse el movimiento, debe controlarse que se haya alcanzado el objetivo con la mano abierta y que luego esta se cierre dentro del margen de acción del objeto. Por último este tipo de objetivo debe tener un estado de desaparición vinculado al objetivo que se use como zona, de tal forma que cuando el estado de este sea fijo (se lleva cabo una transición entre la mano cerrada a mano abierta) y su ubicación esté dentro de la zona objetivo este desaparezca y el usuario obtenga la puntuación correspondiente.

Debe definirse también un tipo especial de objetivo que llamaremos “Zonal”, este tipo de objetivo será el usado como zona donde depositar el resto de objetivos del ejercicio. Su rango debe ser ligeramente mayor que el de otro objetivo para permitir cierto margen de error al usuario. Este tipo de objetivo sería especial y debe ser ignorado como meta para completar el ejercicio, tal y como son ahora los distractores cuya eliminación no se contempla para completar el ejercicio. En principio solo se podrá definir una zona a la vez para el ejercicio, debe limitarse también sus estados, ya que no debe ser eliminado o arrastrado. Cuando un objetivo “Movable” cambie su estado de móvil a fijo dentro del margen del objetivo de tipo “Zonal”, el primero se desechará.

Como restricción adicional en el editor, ha de escogerse un objetivo “Zonal” antes de poder seleccionar los objetivos “Movibles”. Después de seleccionar y ubicar el objetivo “Zonal” en el Designer, deben bloquearse la selección de todos los tipos de objetivos (normales y distractores) a excepción de los “Movibles”. De esta forma el ejercicio queda acotado y se evitan fallos en el planteamiento.

Para definir las reglas del ejercicio, se usarían aquellas ya definidas en Tango:h, como la duración del ejercicio que ya está implementada o la meta de hacer desaparecer “X” cantidad de objetivos. La única diferencia radicaría en la forma de hacerlos desaparecer, actualmente desaparecen cuando los “tocas”, con estos nuevos tipos de objetivos desaparecerían al entrar en contacto con el objetivo “Zonal” y su estado (el de los “Movibles”) fuese fijo.

- **Secuencias de abrir y cerrar manos:** El objetivo de este tipo de ejercicio sería el de posicionar la mano abierta o cerrada en una secuencia determinada cambiando de posición para cada objetivo. Así se obligaría al movimiento de los brazos y a memorizar la secuencia de manos a abiertas y cerradas. El número a memorizar dependería de la dificultad que quisiese aplicarse al jugador.

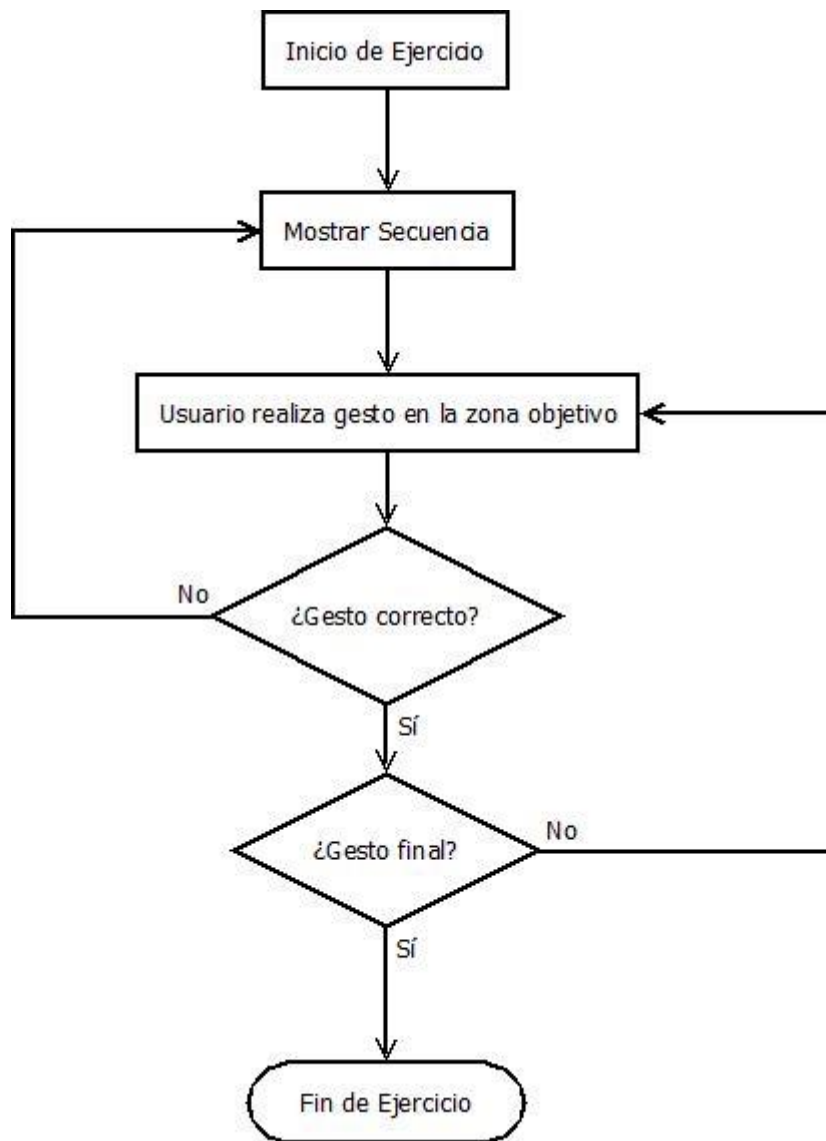


Figura 17. Diagrama de ejercicio de secuencias.

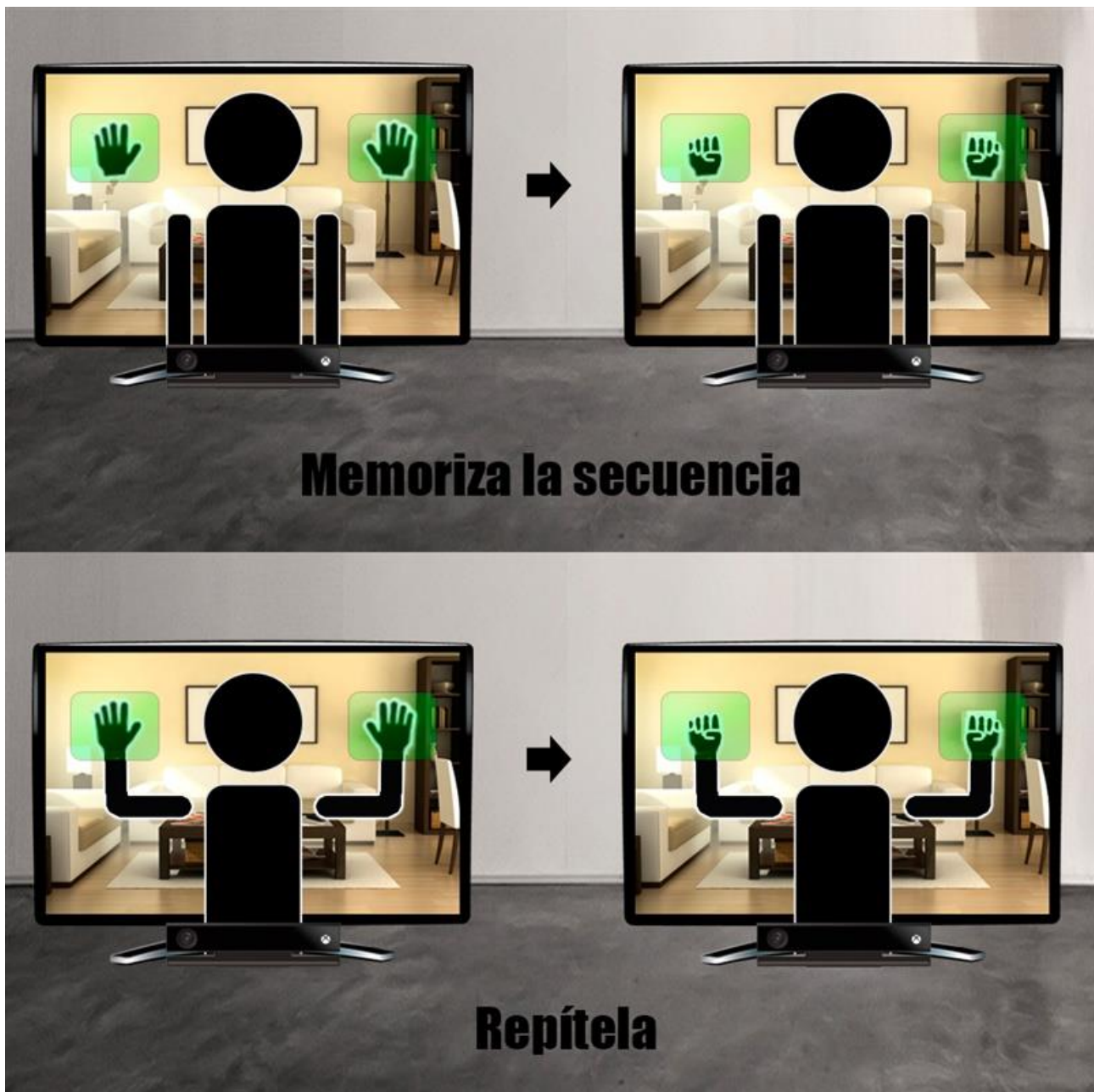


Figura 18. Ejemplo de secuencias de manos.

Diseño de la implementación:

Actualmente Tango:h Designer cuenta con tres opciones que definen el tipo de ejercicio que se va a crear. Para crear un nuevo tipo de ejercicio como el de las secuencias de manos hará falta una remodelación del aspecto visual del Designer que se ubicaría en una cuarta opción que llamaremos “Secuencias”.

Cuando seleccionemos la opción de “Secuencias” el aspecto será similar al de resto de opciones pero con varios cambios. El primero de ellos será un área

definida dentro de la zona de arrastre del editor de objetivos donde deberá ubicarse la secuencia que realizará el paciente. El número de objetivos de la secuencia se limitará por una variable numérica entera que tendrá un valor mínimo de 3 objetivos y un máximo de 10 (parámetros que podrán modificarse posteriormente). El usuario que crea el ejercicio deberá arrastrar los objetivos, limitados a símbolos de mano abierta o cerrada en la zona de la secuencia hasta el máximo que haya definido para la variable límite.

Cada objetivo tendrá un número que identificará su orden, el que se coloque el primero tendrá un 1 en su "ID", el número se irá incrementando de tal forma que no pueda haber dos objetivos con el mismo número y que estos sean correlativos sin saltos.

Una vez colocada la secuencia, se usará una variable de tiempo medida en segundos para determinar cuánto durará la visualización de dicha secuencia. Cuando el paciente acceda al ejercicio debe aparecer la secuencia completa de manos abiertas y cerradas durante el tiempo marcado por la variable de tiempo citada.

El usuario que maneja el editor debe colocar los objetivos después de definir la secuencia. Debe limitarse el número de objetivos aplicables al número de objetivos en la secuencia y se les asignarán número, el número y el estado deben coincidir con el objetivo de misma ID perteneciente a la secuencia para que se añada con éxito al ejercicio.

Una vez ejecutado el ejercicio y mostrada la parte de la secuencia al usuario irán apareciendo uno a uno los objetivos correspondientes en orden ascendente según la secuencia. Los objetivos serán normales, la diferencia radicará en la forma de "eliminar" dicho objetivo. El método usado por Tango:h es el de detectar que la articulación correcta "toca" el objetivo, en este caso sería igual pero añadiendo el estado de la mano, abierta o cerrada en función del tipo de objetivo que se haya definido. Los objetivos con manos cerradas deberán desaparecer cuando una mano en estado cerrado les toque y lo mismo pero al contrario con los objetivos de manos abiertas.

Para establecer las reglas generales puede usarse el sistema de tiempo general para el ejercicio y el resto de opciones. Solo se limitaran las articulaciones (únicamente las manos serán seleccionables) y los objetivos, que

serán el tipo especial de objetivos de “secuencia” de manos abiertas y cerradas.

- **Añadido a los ejercicios actuales:** Puede añadirse otra capa de complejidad a los ejercicios consistentes en alcanzar los objetivos que ya existen en Tango:h. Con esta mejora los ejercicios podrían requerir que algunos ejercicios específicos se alcanzase con la mano abierta o cerrada haciendo uso de los diferentes tipos de objetivos definidos en los anteriores apartados.

4.2.2 El estado Lasso.

Haciendo uso del estado lasso existen varias ideas que pueden llegar a materializarse en nuevos tipos de ejercicios. A continuación se listan algunos planteamientos para nuevos tipos de ejercicios:

- **Rodear objetivos específicos:** En el juego más básico de Tango:h se tiene por meta tocar los objetivos. Con esta nueva función de dibujar áreas se puede pedir al usuario que dibuje un área para seleccionar los objetivos. Esto añade ejercicio adicional y permite que no se limite las reacciones a simples estiramientos de las extremidades.

Diseño de la implementación:

El cambio que debe realizarse consiste en añadir un tipo nuevo de objetivo que pasaremos a llamar “Rodeo”. Los objetivos de rodeo funcionarían igual que los objetivos estáticos de Tango:h pero cambiando su forma de ser “obtenidos” o “eliminados”. Poniendo la mano en modo lasso el dispositivo Kinect puede detectarla como si se tratase de un “puntero”, se usarían entonces las coordenadas para calcular el área dibujada y determinar si dichas

coordenadas obtenidas coinciden con las de un objetivo, en caso positivo este se eliminaría, se daría por marcado. La elección de los objetivos “Rodeo” debe limitar las articulaciones a las manos como únicas seleccionables.

Para el resto de reglas y restricciones del ejercicio pueden aplicarse las que ya están incorporadas a Tango:h ya que la finalidad básica de los ejercicios se mantendrían inalteradas con esta propuesta.

- **Unir objetos:** Se trataría de un ejercicio cuyo objetivo consistiría en que el jugador trazase una línea entre todos los objetivos. Habría un tiempo límite para realizar el ejercicio y para ganar el usuario deberá conectarlos todos. Si el usuario que crea el ejercicio lo desea, puede enlazarlo con otros tipos de objetivos normales.



Figura 19. Ejemplo de unir objetos.

Diseño de la implementación:

Los objetivos de unir objetos a los que llamaremos a partir de ahora objetivos de “Unión”, serían un nuevo tipo de objetivos muy similares a los estáticos actuales pero añadiendo un estado de marca. Haciendo uso de las coordenadas proporcionadas por el estado lasso de la mano y devueltas por el SDK de Kinect, los objetivos se irían marcando.

Para marcarlos todos el estado lasso debe mantenerse en todo momento, si la mano no estuviese en lasso se deberían comprobar todos los objetivos y volver a dejarlos sin “marcar”. Cuando todos los objetivos estén “marcados” o el tiempo llegue a su límite el ejercicio terminaría. Los objetivos no desaparecerían hasta haberlos marcados a todos.

Las reglas generales ya definidas en Tango:h como la duración del ejercicio o el rango de los objetivos se aplicarían de la misma forma. El Designer albergaría el nuevo tipo de objetivo, que podrá tener cualquier imagen asociada y cuyas únicas articulaciones posibles seleccionables en el editor deben ser las manos.

4.3 Prototipo de detección de mano.

Para garantizar que las nuevas funciones que aporta Kinect son alcanzables en un futuro para el proyecto se ha decidido realizar un prototipo que nos ayude a realizar varias pruebas con la función de la detección de manos.

El objetivo de las pruebas es medir la posibilidad y la eficacia de Kinect 2.0 a la hora de detectar los diferentes estados de las manos.

4.3.1 Pruebas de caso de estudio.

Usando un código desarrollado para detectar los diferentes estados de la mano se llevaron a cabo varias pruebas para ver con que fiabilidad y rapidez podía llegar a responder el dispositivo.

El tiempo de respuesta del dispositivo es muy rápido, incluso alternando diferentes estados en las dos manos simultáneamente. La detección es más precisa si las palmas de la mano están mirando hacia Kinect. Si las palmas están dirigidas al suelo o hacia el techo es más propenso a fallar la identificación del estado.

La distancia también afecta al hardware, que es capaz de dar una fiabilidad absoluta en rangos de dos a un metro. Si nos separamos hasta una distancia de tres metros el índice de fallos en la detección aumenta. Sobre todo el reconocimiento del estado Lasso cuyo índice de aciertos a tres metros es prácticamente nulo debido a que el aparato necesita de un objetivo lo suficientemente grande para poder entrar en estado lasso. De hecho y según la

documentación de la propia Microsoft, el estado lasso se puede lograr solo con el dedo índice si el objetivo está lo suficientemente cerca. Por tanto el uso del dedo corazón tiene como objetivo hacer más grande la postura para reconocer el estado lasso.

4.3.2 Pruebas de profundidad.

Se realizaron varias pruebas de profundidad comparando los sensores Kinect 1 y Kinect 2.

Para la primera prueba se colocó un objeto (una caja) sobre una superficie plana y ligeramente inclinado de tal forma que la mitad derecha del objeto tuviese algo más de profundidad que la parte izquierda. Luego usamos un programa para medir la profundidad que marca el dispositivo, este fue colocado a una distancia de un metro aproximadamente. En las siguientes imágenes puede verse el valor de la profundidad (Depth) medido en milímetros. La imagen izquierda (en tonos más azules y verdes) pertenece a Kinect 1 y la de su derecha (en tonos más anaranjados) pertenece a Kinect 2.

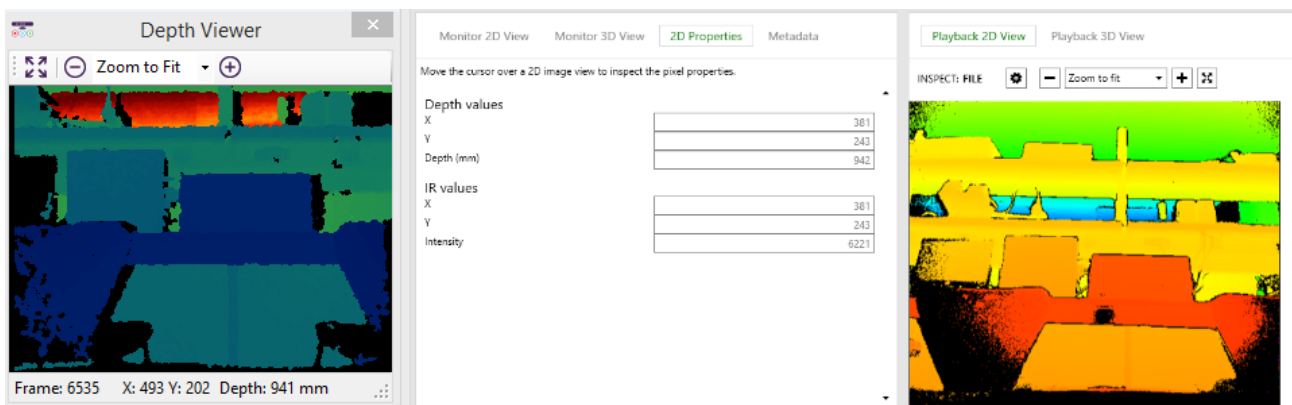


Figura 20. Profundidad de objeto por la derecha.

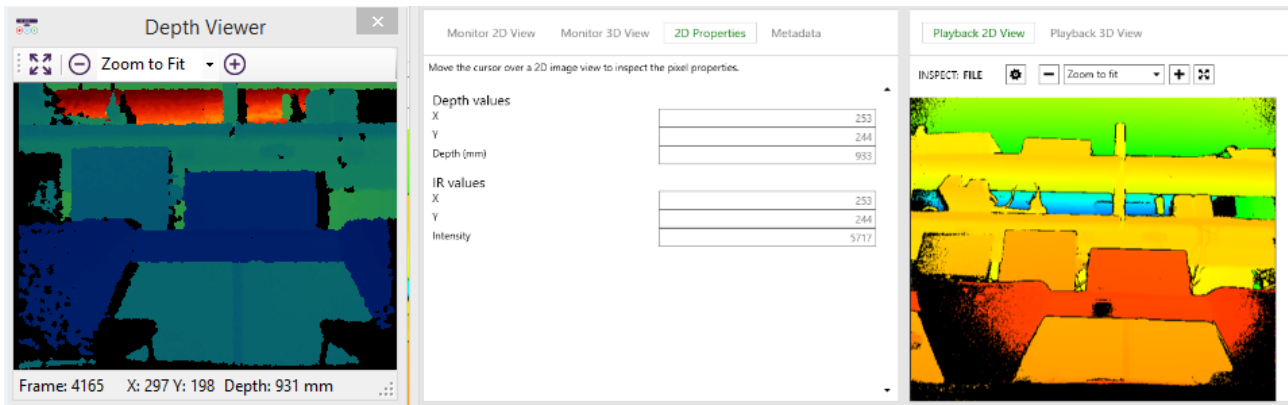


Figura 21. Profundidad de objeto por la izquierda.

En las imágenes se aprecia una ligera diferencia que puede deberse al cambio de tamaño y ángulo inherente a ambos modelos. Pero al realizar ambas pruebas puede apreciarse un peor rendimiento en Kinect 1 ya que para calcular la profundidad de un solo punto la lectura en muchas ocasiones no es clara, oscilando intermitentemente entre dos valores incluso con el puntero totalmente quieto.

También se observa una mayor aparición de zonas negras, estas zonas representan los puntos cuya profundidad no se puede medir.

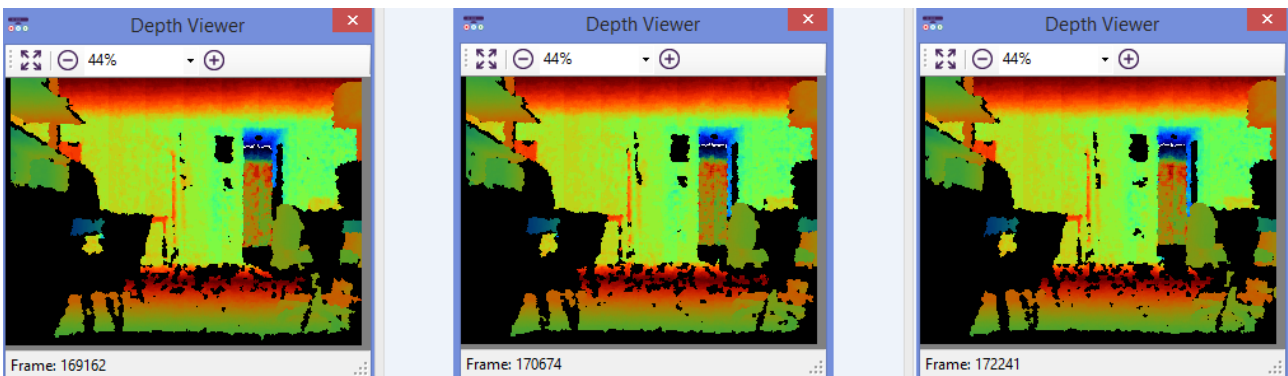


Figura 22. Prueba profundidad Kinect 1.

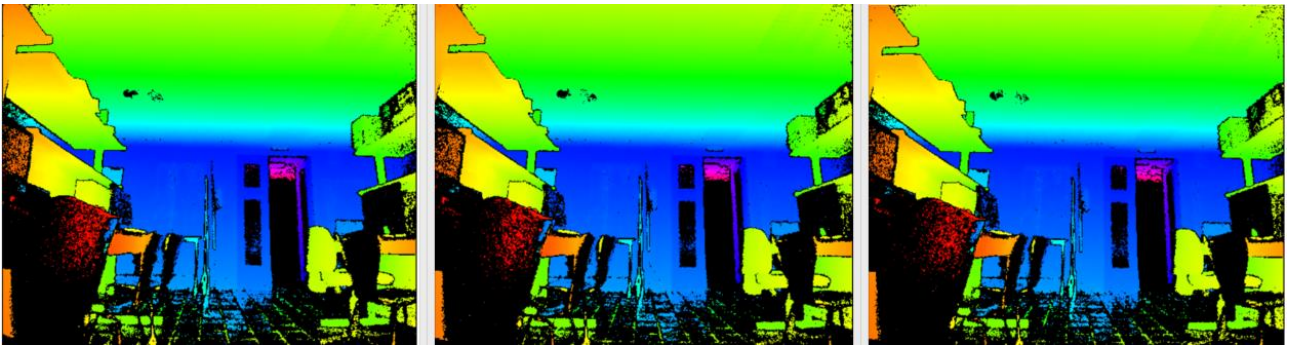


Figura 23. Prueba profundidad Kinect 2.

La comparativa se ha hecho con 3 frames distintos del mismo escenario. En Kinect 2 encontramos un mayor detalle en la imagen de profundidad, podemos observar como las zonas negras son menores e incluso se pueden distinguir las baldosas del suelo mientras que en la primera Kinect las formas se difuminan y no son claras. Con Kinect 2 se aprecian parte de las sillas mientras que con Kinect 1 prácticamente todas son negras lo que implica que no es posible leer su profundidad. Por último, podemos apreciar una mayor variabilidad de las zonas en negro en los frames de Kinect 1, esto hace que muchos valores que se aportan a un frame se pierdan al siguiente y por tanto que la lectura de profundidad no sea tan estable como con Kinect 2.

Estas pruebas de profundidad han servido en definitiva para aclarar algunas de las muchas mejoras de rendimiento que aporta Kinect 2 por encima de su anterior versión de hardware.

Capítulo 5. Incorporación de información facial y emocional.

Kinect permite aplicar reconocimiento facial de forma sencilla haciendo uso de su propio SDK. Una de las características que hacen a esta tecnología tan interesante es la capacidad de detección de la cara hasta un margen de unos 3 metros y medio, lo cual implica que no hace falta estar muy cerca de la cámara para que el reconocimiento sea efectivo. Esto le da una mayor versatilidad y utilidad a la hora de implementarlo en una plataforma de juegos donde no es cómodo que los usuarios tengan que acercarse al sensor todo el rato para detectar sus caras.

Según la documentación de la propia Microsoft Kinect recoge varios tipos de información de la cara que vamos a analizar a continuación para comprender a grandes rasgos, cómo funciona el dispositivo.

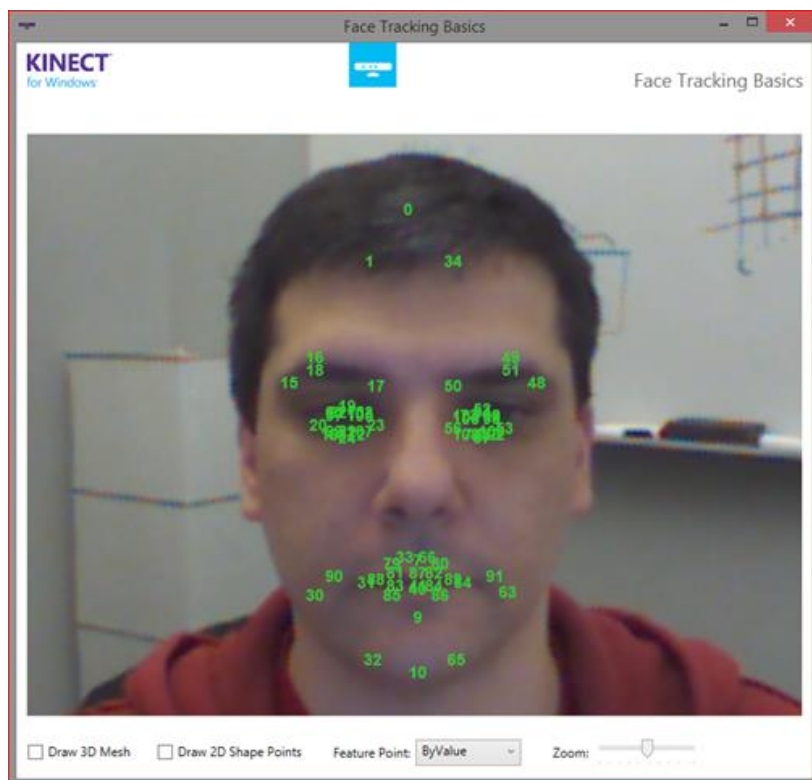


Figura 24. Reconocimiento facial con Kinect.

Kinect detecta la cara de cada jugador y la enmarca en un recuadro para hacerle el seguimiento. Dentro del área el hardware detecta 5 puntos: ojo izquierdo, ojo derecho, nariz y las esquinas izquierda y derecha de la boca. También puede calcularse la rotación de la cara medida en metros.

La variable más importante que debemos tener en cuenta es aquella que recoge las propiedades de la cara. A continuación se ha diseñado una tabla que indica las propiedades básicas que pueden usarse a través del sensor.

Propiedades	Descripción
Feliz (Happy)	El usuario parece estar sonriendo. Es capaz de detectar la sonrisa de las risas. Pero hay que tener en cuenta que un usuario podría sonreír sin estar necesariamente feliz o contento.
Atento (Engaged)	Combina varias propiedades para saber si el jugador está prestando atención al ejercicio o no está centrado en él.

Lleva Gafas (WearingGlasses)	El usuario lleva gafas.
Ojo Izquierdo cerrado (LeftEyeClosed)	El usuario tiene el ojo izquierdo cerrado.
Ojo derecho cerrado (RightEyeClosed)	El usuario tiene el ojo derecho cerrado.
Boca abierta (MouthOpen)	El usuario tiene la boca abierta.
Boca en movimiento (MouthMoving)	El usuario está moviendo la boca. Para detectar mejor esta propiedad el usuario debe quedarse quieto.
No mira (LookingAway)	El usuario no está mirando a la pantalla. Esta propiedad puede no ser precisa, detectando solo movimientos bruscos de la cabeza

Tabla 3. Propiedades de la cara para Kinect

Para cada propiedad Kinect guarda diferentes valores, de tal forma que estos valores pueden ser leídos para saber si se aplica dicha propiedad o no. La siguiente tabla representa los valores que pueden llegar a tener cada una de las propiedades anteriormente citadas.

Resultados de la detección	Descripción
Sí (Yes)	Se tiene una elevada certeza de que la propiedad es positiva.
No (No)	Se tiene una elevada certeza de que la propiedad es negativa.
Quizás (Maybe)	Se cree que la propiedad es positiva. La certeza no es tan elevada como con el Sí.
Desconocido (Unknown)	No se tiene la suficiente información para determinar el estado. Puede deberse a múltiples factores, desde poca visibilidad a un movimiento elevado del usuario.

Tabla 4. Valores de las propiedades de la cara.

Centrándonos ahora en la parte de las emociones, estas son reacciones psicofisiológicas que representan modos de adaptación a ciertos estímulos de

un individuo cuando este percibe un objeto, persona, lugar, suceso, o recuerdo importante. En Tango:h se busca modificar un sistema recomendador de ejercicios para que sea usado en una elección inteligente de las tareas a realizar para la rehabilitación del paciente. Por ejemplo, si el paciente expresa descontento o enfado al perder en un ejercicio, el recomendador puede ajustar la próxima prueba para que esta sea más fácil con el fin de que el jugador no reciba demasiado estrés o frustración.

También puede tornarse en una importante herramienta de feedback para el especialista encargado de la rehabilitación. Ya que podría consultar en la pantalla de progresión del usuario las emociones que tiene un paciente con cada tipo de ejercicio que realiza y obrar en consecuencia.

Una vez analizada la forma en la que Kinect desempeña el reconocimiento facial y tenemos en cuenta lo que implica una emoción, nos percatamos de que para llevar a cabo un reconocimiento emocional del jugador son necesarias herramientas que nos permitan determinar estados más complejos.

Por ello se nombran a continuación algunos de los trabajos y herramientas más importantes en este campo con el fin de determinar la orientación que debe o puede aplicarse al proyecto de Tango:h a este respecto.

No son pocas las empresas e instituciones actuales que investigan la aplicación del reconocimiento facial pero son muy pocas las que liberan el trabajo que están realizando. Por mencionar uno de los sistemas, está el del Instituto de Sistemas Integrados Fraunhofer (Alemania) que está desarrollando un software capaz de “leer” el rostro. Calificado como un “lector de emociones”.

La clave del proyecto estaría en una librería llamada SHORE de código propietario y desarrollada por el propio Instituto Alemán. El software es capaz de diferenciar 4 tipos de emociones entre felicidad, enfado, tristeza y sorpresa. Además es capaz de diferenciar el sexo de la persona y de determinar su edad con un margen de hasta 7 años de error.

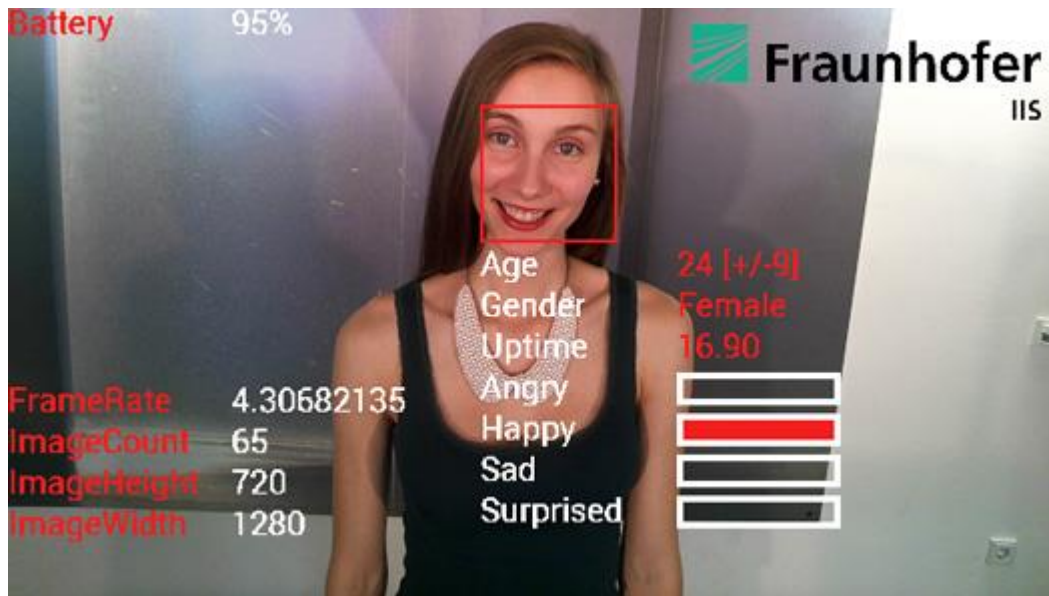


Figura 25. Programa SHORE en funcionamiento.

El mayor problema se encuentra en el código propietario ya que no es un material de uso libre. Además el prototipo funciona actualmente solo con las Google Glass, aunque desde el Instituto se asegura que serviría para cualquier cámara y por tanto también para Kinect en un futuro.

En la Universidad Pontificada de Comillas en Madrid se desarrolló un software que funciona directamente sobre Kinect para reconocimiento facial. De dicho proyecto se pueden extraer las partes más importantes a la hora de plantearse una aplicación similar. La solución consta de tres partes que son:

- **Una herramienta de adquisición de las imágenes faciales que haga uso de Kinect.** La propia universidad creó su propia herramienta usando el lenguaje C#. Esto implicaría una alta posibilidad de compatibilidad con Tango:h, así como la posibilidad de crear nuestra propia herramienta desde cero usando el mismo lenguaje.
- **Una herramienta de procesamiento de las imágenes faciales.** Para la parte de procesamiento de imágenes es necesaria una herramienta realmente potente en el ámbito matemático y de gestión de grandes cantidades de datos. Por eso en el citado proyecto se hizo uso de una herramienta tan potente como Matlab.

- **Y una herramienta de análisis de las imágenes faciales.** Esta parte consiste principalmente en un programa que muestre los resultados obtenidos. En nuestro caso los resultados obtenidos deberán ser usados directamente por Tango:h y no es necesario mostrarlos por pantalla si no se requiere. Bastaría simplemente con que el recomendador los obtuviese y decidiese en consecuencia.

El método más usado cuando se desarrollan este tipo de aplicaciones es el de la comparación. El software captura la cara del usuario y la pasa a la herramienta analítica, esta se encarga de determinar los rasgos y compara los datos obtenidos de la cara con todas las caras que tiene almacenadas en su base de datos. Cuando encuentra una lo suficientemente similar se revisa a que emoción pertenece y se le asigna al usuario la misma emoción.

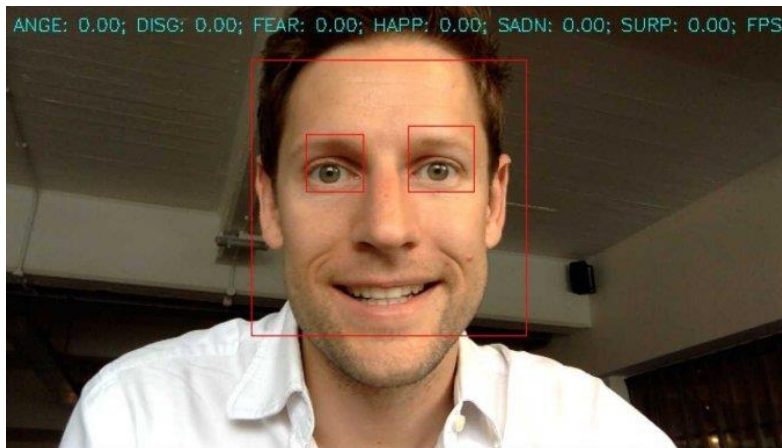


Figura 26. Reconocimiento emocional con CrowdEmotion.

Como se ha dicho al principio de este apartado, el significado del mismo es la búsqueda de unas líneas claves básicas de cara a un futuro proyecto, comprendiendo el funcionamiento actual de Kinect 2.0 y del estado actual de algunas de las investigaciones que se han o están llevándose a cabo. Pudiendo ser de utilidad para agregar sistema recomendador basado en emociones en Tango:h.

5.1 Propuestas

Una de las mayores utilidades para las que puede aplicarse el reconocimiento facial simplificado que ofrece Kinect 2.0 es en la rama de rehabilitación orientada a la parálisis facial.

La parálisis facial es una lesión ubicada en el nervio facial, esta afección a menudo causa síntomas como la ausencia de movilidad en los músculos de la cara, ya sea del lado izquierdo o del lado derecho. También puede presentar la incapacidad para cerrar un ojo o la desviación de la boca entre otros síntomas.

Existen muchas terapias y ejercicios para tratar la lesión y algunos de los citados pueden reproducirse gracias a la detección de Kinect. Los que más nos interesan son los denominados “ejercicios activos”. La siguiente tabla muestra los ejercicios y la propiedad perteneciente a Kinect que puede usarse para realizarlos.

Ejercicio	Propiedad de Kinect
Sonrisa amplia	Puede usarse la propiedad Feliz (Happy) para determinar si la persona está sonriendo o no.
Dilatar el ala de la nariz hasta levantar el labio superior y enseñar los dientes.	Para este ejercicio pueden usarse la propiedad de boca abierta (MouthOpen) en sus diferentes estados para determinar si está cerrada o no.
Cerrar y abrir los ojos.	Puede usarse la propiedad de los diferentes ojos cerrados para determinar la realización del ejercicio.
Contraer los labios como si se fuera a dar un beso.	La propiedad de boca en movimiento (MouthMoving) puede servir para diseñar un ejercicio que sirva a tal efecto..
Apretar fuertemente los ojos.	Para este ejercicio pueden usarse las propiedades para cerrar el ojo derecho e izquierdo de forma simultánea. También habría que añadir un aviso de tiempo para informar al paciente de que tiene que volver a abrirlos.

Tabla 5. Ejercicios rehabilitación facial.

Capítulo 6.

Conclusiones y líneas futuras

Al inicio del proyecto me encontré con una herramienta, Tango:h, que poseía y posee un potencial enorme. La importancia de su aplicación en el campo de la rehabilitación con niños es impresionante.

Pero su aplicación no se limita solo a los más pequeños, también es posible que los adultos con algún tipo de problema puedan recuperarse poco a poco haciendo uso de ella. Las mejoras que puede suponer para un paciente tanto el aspecto físico (movilidad, mejora de respuesta motriz, etc.) como en el aspecto psicológico (reflejos mentales, capacidad para memorizar, etc.) son impresionantes.

Y el potencial de la herramienta no solo tiene que ver con los pacientes. Tango:h puede mostrar la evolución en su rendimiento al especialista a cargo de su tratamiento. Unas estadísticas muy útiles tanto a nivel individual como colectivo, ya que puede aportar datos contrastados en estudios de tratamientos sobre un número determinado de pacientes que posean el mismo tipo de problema.

Sin olvidar por supuesto que a diferencia de una herramienta que posea una capacidad fija, la capacidad de Tango:h es prácticamente ilimitada al estar dotada de un editor de ejercicios que ofrece una amplia gama de posibilidades. Adaptando los ejercicios a casos concretos, haciéndolos aún más efectivos.

Centrándonos ahora en el trabajo realizado, la inclusión del uso de Kinect 2.0 en Tango:h se ha realizado no sin percances o problemas de diversa índole. Y aunque el tiempo invertido en analizar las líneas y líneas de código que componen ambas herramientas (Tango:h/Tango:h Designer y Advant) ha sido mayor del esperado, se puede decir que se han cumplido los principales requisitos del trabajo. Hemos incluso llegado a analizar varios de los cambios que supone la inclusión del nuevo dispositivo y diseñado varios posibles ejercicios.

En cuanto al futuro del proyecto hay un par de ramas que tratar. De manera interna al proyecto todavía quedan cosas por pulir. Se debe buscar una solución viable para volver a incluir la función que ha debido eliminarse en el transcurso de este proyecto. Por otra parte espero que la idea de los ejercicios nuevos incluidos en la investigación realizada en este trabajo, puedan ser aplicados en un futuro en la plataforma y sirvan de ayuda al próximo equipo que se encargue de la herramienta.

Si hablamos del futuro más externo del proyecto, la verdad es que sería conveniente que se diese a conocer aún más esta magnífica herramienta. Ahora mismo existe la Asociación Down Tenerife y hay varios hospitales que han probado y criticado positivamente a Tango:h. La idea es hacerla llegar al mayor número de instituciones y pacientes que la necesiten. Aunque quizás la mayor desventaja de Kinect 2.0 con respecto a su versión anterior sea el encarecimiento del dispositivo que ya no estaría catalogado como de “bajo coste”. Aun así considero que la inversión bien vale la pena.

En lo personal el trabajo me ha permitido explorar un nuevo lenguaje, el C# y un nuevo entorno el Visual Studio y el Framework .NET en su última versión. Estoy seguro de que toda la experiencia obtenida con estas herramientas dará sus frutos en un futuro y serán un peldaño más en mi realización profesional como Graduado en Ingeniería Informática.

Capítulo 7.

Summary and Conclusions

When the Project began I found Tango:h, a tool that had an enormous potential. The importance of this software in the field of rehabilitation with children is impressive.

But its application is not limited to the children; it is also possible for adults with some kind of problem to recover gradually using it. Tango:h can greatly improve physical and psychological aspect of the patient, like its movability, mental reflexes and ability to memorize.

And the potential of the tool not only has to do with patients. Tango:h can show trends in performance to the specialist in charge of your treatment. The statistics are useful both individually and collectively, as it can provide hard data on treatment studies on a number of patients who have the same kind of problem.

Not forgetting that unlike a tool that has a fixed capacity, the ability to Tango:h is virtually unlimited to be provided with an exercise editor that offers a wide range of possibilities. Adapting exercises to specific cases, making them even more effective.

Turning now to the work done, we had some problems to include the Kinect 2.0 in Tango:h. The time spent on analyzing the lines of code has been larger than expected, but we can say now that we have met the main requirements of the job. We've even come to analyze several of the changes caused by the inclusion of the new device and designed several possible exercises.

As for the future of the project are a couple of branches to deal with. Internally the project has still things to polish. We must find a viable solution to reinstate the function that had to be removed in the course of this project solution.

Furthermore I hope that the idea of the new exercises included in the research in this work can be applied in the future on the platform and will assist the next team.

If we talk about the outermost future of the project, now the “Asociación Down Tenerife” (Tenerife, Spain) is using Tango:h and we have several hospitals that have tested and positively criticized the software. The idea is to make it reach the largest number of institutions and patients who need it. But perhaps the biggest disadvantage of Kinect 2.0 with respect to the previous version of the device is that the increase in price would no longer be classified as “low cost”. Yet I believe that the investment is well worth it.

Personally the work has allowed me to explore a new language (C#), the Visual Studio environment and the .NET Framework in its latest version. I am sure that the whole experience with these tools will pay off in the future and will be one more step in my professional achievement as a graduate in computer engineering.

Capítulo 8.

Presupuesto

Para el presupuesto se han tenido en cuenta las horas invertidas según lo estipulado en el anteproyecto sobre los objetivos marcados y los cumplidos. También se han tenido en cuenta los materiales necesarios para realizar el Trabajo de fin de Grado. Al ser un dispositivo nuevo, el ordenador necesario debe cumplir ciertos requisitos mínimos y al precio del dispositivo se ha sumado el precio del adaptador correspondiente para su uso en PC.

Tipos de gastos	Importe en euros
Salario en función de las horas invertidas	5019 €
Equipo informático con las especificaciones mínimas de hardware.	700 €
Dispositivo Kinect 2.0	149.99 €
Adaptador Kinect para Windows	49.99 €
Total de presupuesto	5918.98 €

Tabla 6. Tabla de presupuesto.

Bibliografía

- [1] Kinect for Windows SDK 2.0 (2015). <https://www.microsoft.com/en-us/download/details.aspx?id=44561> .
- [2] Technical documentation and tools (2015). <https://www.microsoft.com/en-us/kinectforwindows/develop/downloads-docs.aspx> .
- [3] Kinect for Windows features. <https://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx> .
- [4] Set up the hardware. https://www.microsoft.com/en-us/kinectforwindows/purchase/sensor_setup.aspx .
- [5] Kinect for Windows Programming Guide. <https://msdn.microsoft.com/en-us/library/dn782037.aspx> .
- [6] ITER. (2012). *Manual de usuario*. (2012). <http://tangoh.iter.es/pdf/TangoH.pdf> .
- [7] ITER. (2012) *Manual del diseñador*. <http://tangoh.iter.es/pdf/TangoH%20designer.pdf>
- [8] Microsoft Visual Studio (2015). <http://www.visualstudio.com/>
- [9] Marçal Montserrat (2014). *Características Kinect 2*. <http://www.kinectfordevelopers.com/es/2014/01/28/caracteristicas-kinect-2/>
- [10] Laura Frías Carrillo (2014). *Desarrolla Apps con Kinect v2*. http://sg.com.mx/revista/45/desarrolla-apps-kinect-v2#.VdqHffl_Okq
- [11] Álvaro Ibáñez ‘Alvy’ (2014). *El software que lee las emociones con solo ver nuestras caras*. <http://www.rtve.es/noticias/20141027/software-lee-emociones-solo-ver-nuestras-caras/1036726.shtml>
- [12] Rodrigo Ruiz Martín. *Reconocimiento de expresiones faciales mediante imágenes adquiridas con cámara Kinect*. <http://www.iit.upcomillas.es/pfc/resumenes/540711e37af4c.pdf>

- [13] Maleny Abrego (2013). *Kinect 2.0*.
<https://malenyabrego.wordpress.com/2013/07/13/kinect-2-0/>
- [14] Terapia-Física. *Parálisis Facial*. <http://www.terapia-fisica.com/paralisis-facial.html>