



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: Desarrollo del teclado virtual “Mokey” basado en gestos para personas con movilidad reducida: capa de usuario

AUTOR: Carlos Lázaro Basanta

TITULACIÓN: Grado en Ingeniería de Sonido e Imagen

TUTOR: Martina Eckert

DEPARTAMENTO: Teoría de la Señal y Comunicaciones

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: M^ª Luisa Martín Ruiz

TUTOR: Martina Eckert

SECRETARIO: Enrique Rendón Angulo

Fecha de lectura: 6 de Junio de 2017

Calificación:

El Secretario,

RESUMEN

Una Interfaz Natural es aquel programa en el cuál la interacción se realiza sin utilizar sistemas de mando o dispositivos de entrada, sino por medio de movimientos corporales o gestos. El presente proyecto trata sobre el desarrollo de la capa de usuario de Mokey, una Interfaz Natural diseñada y desarrollada para controlar, por medio de movimientos corporales, aplicaciones de ordenador simulando eventos de teclado. La capa de sistema de la Interfaz Natural Mokey es tratada en la memoria del proyecto realizado por Marcos López García [1].

El principal objetivo del desarrollo de Mokey es el de facilitar, por medio de la tecnología, la accesibilidad a aplicaciones y programas de ordenador al gran colectivo de personas con discapacidad o con movilidad reducida. A día de hoy, en una sociedad tan informatizada y tecnológica, es necesario eliminar la exclusión social que sufren aquellas personas con movilidad reducida a la hora de poder utilizar programas de ordenador en su vida cotidiana. Para ello, se crea una Interfaz Natural con el fin de poder utilizar, por medio de movimientos corporales, aquellas aplicaciones que no fueron diseñadas para ese fin. Es esencial que el diseño de Mokey y su interfaz gráfica de configuración sean simples e intuitivos, pero, al mismo tiempo, sea configurable para el mayor número de personas posible.

En esta presente memoria, se realiza un análisis detallado del Estado del Arte y de los antecedentes en relación a la tecnología implicada en el proyecto. Se presta especial atención al hardware y software empleado para captar el movimiento corporal de los usuarios, la cámara Microsoft Kinect, y las herramientas empleadas para el desarrollo de Mokey. Posteriormente, se realiza el análisis de la solución desarrollada, prestando principal atención a los aspectos relativos al diseño de la interfaz gráfica y a todo el desarrollo de funciones diseñadas para buscar la mayor versatilidad del programa.

Por último, con el objetivo de probar el funcionamiento del programa en casos reales, se somete a una batería de pruebas con voluntarios y se realiza un informe de los resultados obtenidos. Además, se realizan pruebas de compatibilidad, de configuración y de rendimiento de las funciones desarrolladas dentro de Mokey comparándolo con otra herramienta similar, FFAST. El objeto de dicha comparación es el de observar las ventajas que Mokey supone frente a otras aplicaciones que no están diseñadas inicialmente para gente con discapacidad. Los resultados de rendimiento se publicaron en el *International Symposium on Consumer Electronics* (ISCE) en 2015 [2] y las pruebas con usuarios junto con una descripción completa del sistema final se publicará en junio de 2017 en la revista internacional *Assistive Technology* [3].

ABSTRACT

A Natural Interface can be defined as a program in which the interaction is performed without using control systems or input devices, but through body movements or gestures. The present project deals with the development of the Mokey user layer, a Natural Interface designed and developed to control, through body movements, computer applications simulating keyboard events. The system layer of the Mokey Natural Interface was addressed in the final degree project developed by Marcos López García [1].

In fact, Mokey's main goal is to allow people with disabilities or mobility problems the access to new technologies. Not only the Natural Interface is created to be used by people with disabilities using body movements or gestures, but also Mokey's design and graphical configuration interface are simple and intuitive. Consequently, adapting the use of Mokey to as many people as possible is another point to take into account. Easy access to applications is a great advantage for people with reduced mobility. Hence, it is necessary to eliminate the social exclusion suffered by those with reduced mobility, allowing them to take advantage of using computer programs in their daily lives.

The aim of this report is to show a detailed analysis of the state of the art and the background related with the technology involved in the project. Particular attention is paid to the hardware and software used to capture users' body movements, the Microsoft Kinect camera, and the tools used to develop Mokey. Regarding the developed solution, it is analyzed considering aspects related to the graphical interface design and the development functions designed to find the greatest versatility of the program.

Finally, the report analyzes the results of the effectiveness of the program based on a test bench applied in actual cases. In addition, compatibility, configuration and performance tests of the functions developed within Mokey are performed comparing them to a similar tool, FFAST. The purpose of this comparison is to observe the advantages Mokey provides compared to other applications that are not initially designed for people with disabilities. The results on performance have been published at the International Symposium on Consumer Electronics (ISCE) in July 2015 [2] and the user tests together with a complete description of the final system will be published in the international journal *Assistive Technology* [3].

ÍNDICE

Capítulo 1. INTRODUCCIÓN.....	8
1.1. Motivaciones.....	8
1.2. Objetivos.....	9
1.3. Estructura.....	10
Capítulo 2. ANTECEDENTES.....	11
2.1. La discapacidad como problema social.....	11
2.2. La tecnología como posible solución.....	11
2.3. Microsoft Kinect.....	13
2.4. FAAST.....	17
2.5. Otras tecnologías.....	19
Capítulo 3. DESCRIPCIÓN TÉCNICA DE MOKEY.....	20
3.1. Software y Hardware empleados.....	20
3.1.1. SDK v1.8 (<i>Software Development Kit</i>).....	20
3.1.2. Lenguaje C#.....	21
3.1.3. Microsoft Visual Studio 2013.....	21
3.2. Función y descripción.....	22
3.2.1. Concepto de <i>delay</i>	24
3.2.2. Concepto de <i>displacement</i>	25
3.3. Interfaz gráfica.....	26
3.3.1. Elementos.....	33
3.4. Movimientos configurables.....	36
3.4.1 Modo <i>Wheelchair</i>	43
3.5. Grabación de movimientos.....	44
3.6. Registro de datos.....	46
Capítulo 4. RESULTADOS Y ANÁLISIS.....	48
4.1. Experiencia configuración interfaz gráfica.....	48
4.1.1. Comparación con FAAST.....	49
4.2. Interacción de Mokey con aplicaciones de terceros.....	50
4.3. Pruebas con voluntarios.....	51
4.3.1. Pruebas con personas con discapacidad.....	51
4.3.2. Pruebas con personas sin discapacidad.....	55
4.4. Rendimiento registro de datos.....	57

4.5. Limitaciones del programa	60
Capítulo 5. CONCLUSIONES Y LÍNEAS FUTURAS	61
5.1. Conclusiones	61
5.2. Líneas futuras.....	62
REFERENCIAS	64
Anexo 1. PRESUPUESTO.....	67
Anexo 2. MANUAL DE USUARIO DE MOKEY	68
A2.1. Instalación de Mokey	68
A2.2. Configuración de Mokey	68
A2.3. Grabar movimientos personalizados	75
A2.4. Referencias.....	77

ÍNDICE DE FIGURAS

Figura 2.1: Detección de los puntos en “modo cercano” mediante Kinect for Windows. [18]...	15
Figura 2.2: Estructura de los distintos sensores de Kinect 1.0 y Kinect for Windows. [17].....	15
Figura 2.3: Imagen de profundidad (izq.) e imagen RGB (der.) captada de Kinect. [18].....	16
Figura 2.4: Representación los puntos del esqueleto humano que capta Kinect simultáneamente.	16
Figura 2.5: Apariencia principal de los tres módulos de FAAST	18
Figura 2.6: Ventana de configuración de movimientos en FAAST (arriba), eventos seleccionables para la entrada de cada movimiento (abajo izquierda) y eventos seleccionadas para la salida de cada movimiento (abajo derecha).....	19
Figura 3.1: Asistente para la creación de proyectos dentro de Visual Studio 2013	22
Figura 3.2: Editor de interfaz gráfica de un proyecto de Visual Studio 2013.	22
Figura 3.3: Diagrama de flujo entre interfaz, hilo principal e hilos secundarios.	23
Figura 3.4: Diagrama del funcionamiento de Mokey.....	24
Figura 3.5: Representación de los botones de configuración del retardo en la interfaz gráfica..	25
Figura 3.6: Representación de los botones de configuración del desplazamiento dentro de la interfaz.....	25
Figura 3.7: Diagrama navegación de ventanas de la interfaz gráfica de Mokey.....	26
Figura 3.8: Pantalla de bienvenida y de inicio de sesión de Mokey.....	26
Figura 3.9: Directorio de cada sesión en la raíz del disco dentro de la carpeta Mokey.	27
Figura 3.10: Pantalla de carga de configuración dentro de la sesión personal de Mokey.	28
Figura 3.11: Pantalla de edición de datos personales de la sesión personal de Mokey.....	28
Figura 3.12: Pantalla principal de configuración de Mokey.	30
Figura 3.13: Pantalla principal de configuración de movimientos de grabación de Mokey.	32
Figura 3.14: Pantalla de fin de sesión y guardado de Mokey.....	33
Figura 3.15: Estado de reposo del usuario del cual parten todos los movimientos configurables, tanto de pie como sentado.	37
Figura 3.16: Representación gráfica del movimiento 1: Brazo derecho hacia arriba (izquierda) y movimiento 2: Brazo izquierdo hacia arriba (derecha).	38
Figura 3.17: Representación gráfica del movimiento 3: Tronco hacia la derecha (derecha) y movimiento 4: Tronco hacia la izquierda (izquierda).	39
Figura 3.18: Representación gráfica del movimiento 5: Brazo derecho hacia la derecha (derecha) y movimiento 6: Brazo izquierdo hacia la izquierda (izquierda).	40
Figura 3.19: Representación gráfica del movimiento 7: Pie derecho hacia la derecha (derecha) y movimiento 8: Pie izquierdo hacia la izquierda (izquierda).....	41

Figura 3.20: Representación gráfica del movimiento 9: Pie derecho hacia delante (izquierda) y movimiento 10: Pie izquierdo hacia delante (derecha).	42
Figura 3.21: Representación gráfica del movimiento 11: Brazo derecho hacia atrás (izquierda) y movimiento 12: Brazo izquierdo hacia atrás (derecha).	43
Figura 3.22: Control de activación “Modo Wheelchair” en la pantalla principal de Mokey.	43
Figura 3.23: Módulo movimiento grabado de la ventana principal de Mokey	45
Figura 4.1: Gráfico del registro de datos con el equipo 1 y muestreo de 0,5 segundos	58
Figura 4.2: Gráfico del registro de datos con el equipo 2 y muestreo de 0,5 segundos	59
Figura 4.3: Gráfico del registro de datos con el equipo 2 y muestreo de 0,2 segundos	59
Figura A2.1: Pantalla de bienvenida e inicio de sesión de Mokey.....	69
Figura A2.2: Pantalla de selección de archivo de configuración de Mokey	69
Figura A2.3: Pestaña de introducción de datos personales de la sesión de Mokey.....	70
Figura A2.4: Mensaje de advertencia del programa al no detectar Kinect correctamente.	70
Figura A2.5: Ventana de advertencia del programa al no detectar Kinect correctamente.	71
Figura A2.6: Selección modo Wheelchair en la ventana principal de configuración	71
Figura A2.7: Selección de movimientos y ajuste del parámetro displacement.	72
Figura A2.8: Selección de tecla asociada a cada movimiento y ajuste del parámetro delay.....	73
Figura A2.9: Desplegable de la selección de tecla asociada a cada movimiento.	73
Figura A2.10: Módulo relativo a la aplicación controlada con Mokey.....	74
Figura A2.11: Ejemplo de título de ventana de un archivo de PowerPoint 2016.	74
Figura A2.12: Botón relativo a la carga de un archivo de configuración de la interfaz gráfica..	74
Figura A2.13: Ventana de guardado de archivos de configuración.	75
Figura A2.14: Ventana de configuración de la grabación de movimientos.	76
Figura A2.15: Mensaje informativo al realizar la grabación de un movimiento.....	77

ÍNDICE DE TABLAS

Tabla 2.1: Principales diferencias entre Kinect 1.0 y Kinect for Windows.	14
Tabla 3.1: Relación de movimientos con los huesos implicados y eje de interés	37
Tabla 3.2: Relación de movimientos con los huesos implicados en modo “Wheelchair”.	44
Tabla 3.3: Contenido fichero csv de registros de parámetros de uso	47
Tabla 4.1: Resultados tiempos de configuración para Mokey y FFAST	49
Tabla 4.2: Voluntarios para las pruebas de Mokey	52
Tabla 4.3: Resultados estadísticos del uso de movimientos.....	53
Tabla 4.4: Número de intentos para cada programa.....	53
Tabla 4.5: Movimientos probados por personas sin discapacidad y tasa de éxito	56

Capítulo 1. INTRODUCCIÓN

Este proyecto ha sido resultado del trabajo en grupo junto con Marcos López [1] y realizado en dos etapas diferenciadas. Una primera etapa de seis meses donde se cursan prácticas curriculares en el Centro de Investigación en Tecnologías Software y Sistemas Multimedia para la Sostenibilidad (CITSEM). Durante este periodo se establece el estudio de la tecnología existente en ese momento para posteriormente elaborar una aplicación simple que, por medio de la cámara Microsoft Kinect y el SDK de desarrollo 1.8 de Microsoft, transforma movimientos corporales simples en la pulsación de tecla requerida. La interfaz gráfica es simple al igual que los movimientos que se permiten seleccionar, seis en concreto. En este periodo la prioridad es la de establecer una conexión y un flujo de datos estable para posteriormente continuar el desarrollo de lo que más tarde pasó a denominarse Mokey. En esta etapa ya se introdujeron los parámetros de retardo y amplitud de cada movimiento, pero su ajuste y uso era limitado.

La segunda etapa transcurrió inicialmente durante un periodo de ocho meses con la realización de una Beca de Colaboración en el Departamento de Teoría de la Señal y Comunicaciones de la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETSIST) y posteriormente un periodo de seis meses de trabajo libre. Durante este periodo se realizaron pruebas de rendimiento y pruebas con personas físicas, así como optimización y desarrollo de mejoras en la interfaz e introducción de nuevas funciones como el registro de sesiones, grabación de movimientos y registro de datos de uso.

El proyecto se presenta en dos memorias diferenciadas. La primera memoria hace referencia a los algoritmos, funciones empleadas, rendimiento y el desarrollo de la capa de sistema de Mokey, presentado en [1] por Marcos López García. Al finalizar la segunda etapa y todas las pruebas de uso con personas, se realiza la segunda memoria referente a la capa de usuario de Mokey: interfaz gráfica, experiencia de usuario, registro de sesiones y datos de uso, pruebas con grupos de personas, etc. Esta segunda memoria es la que se refiere a estas líneas.

1.1. Motivaciones

Según los datos del informe de la Organización Mundial de la Salud (OMS) del año 2016 [4], en la población mundial hay más de 1000 millones de personas con algún tipo de discapacidad, dato que se acerca la séptima parte de la población mundial. Cerca de la cuarta parte de este colectivo no tiene acceso a los medios necesarios para depender de sí mismo, necesitando la asistencia de otras personas o recursos institucionales. La sociedad no es consciente de la magnitud de este dato, pasando desapercibido y sin darle la relevancia que en realidad tiene.

La tecnología es el recurso más importante para proporcionar las necesidades que demanda el colectivo de personas con discapacidad. Mediante las herramientas y recursos nacidos por medio de la tecnología, se refuerza la accesibilidad de estas personas a las actividades cotidianas. La Realidad Aumentada busca la convergencia entre la información del mundo real con la información tomada virtualmente por otros dispositivos tecnológicos. Esta convergencia proporciona la información necesaria para crear aplicaciones y herramientas inteligentes con mayores posibilidades. Las Interfaces Naturales son un ejemplo de ello ya que se trata de aplicaciones que, mediante la información de diferente tipo captada por dispositivos electrónicos, son capaces de generar nuevas acciones. Mediante los nuevos recursos que proporciona la Realidad Aumentada, es posible conseguir una mayor accesibilidad e integración de las personas con discapacidad y evitar la segmentación de este colectivo y la exclusión social que ello conlleva. En otras palabras, mediante las nuevas tecnologías es posible eliminar las barreras físicas y sociales de discriminación de este determinado colectivo.

La tecnología, en la actualidad, es algo esencial en la vida de las personas ya que, a día de hoy, es imposible concebir una vida sin teléfono inteligente, tableta, televisión, ordenador, Internet, etc. Para combatir la exclusión social y la accesibilidad a recursos y tareas cotidianas surge Mokey, nombre que proviene de las palabras inglesas *Motion* (Movimiento) y *Key* (tecla). Esta aplicación es una Interfaz Natural que permite el uso de programas de ordenador mediante movimientos corporales del usuario de forma que las personas con movilidad reducida puedan utilizar gran parte de las aplicaciones y juegos que no han podido utilizar correctamente hasta este momento. El dispositivo empleado para captar la información de los movimientos corporales del usuario es la cámara Microsoft Kinect [5].

Además de asistir en ámbitos de accesibilidad a usuarios con discapacidad motora o movilidad reducida, la Interfaz Natural Mokey está orientada a ámbitos de rehabilitación de forma que, mediante juegos o aplicaciones lúdicas, se facilite la tarea de seguimiento de este colectivo por parte de un terapeuta. La solución proporcionada se ajusta a los casos reales, realizando las pruebas oportunas para conseguir un objetivo a la altura de las grandes necesidades de este colectivo.

1.2. Objetivos

El principal objetivo de Mokey es el crear una Interfaz Natural que permita el uso de cualquier aplicación de ordenador, controlable con el teclado, mediante movimientos corporales. Es una aplicación diseñada especialmente para el uso por personas con discapacidad para la realización de ejercicios de rehabilitación. Por ello, se han diseñado una serie de objetivos en el proyecto con el fin de mejorar las herramientas ya existentes.

El objetivo es el desarrollar un *middleware* que, mediante la información de movimiento corporal proporcionada por Kinect, ejecute los eventos de teclado previamente configurados en la interfaz gráfica del programa. La redacción de esta memoria se enmarca dentro de los siguientes objetivos específicos de Mokey:

- Simulación de pulsaciones de teclado mediante 12 movimientos diferentes lo más útiles posible para los diferentes tipos de discapacidad.
- Posibilidad de ajustar el periodo entre pulsaciones de teclado, así como el rango o amplitud del gesto de cada uno de los movimientos configurados para ejecutar el evento de teclado correspondiente.
- Simulación de pulsaciones de teclado mediante 4 movimientos grabados con ambas piernas y brazos permitiendo ajustar igualmente sus parámetros. Este modo podría hacer que el programa se adecúe más a las necesidades de cada usuario.
- Crear un modo especialmente dedicado a aquellas personas que permanecen en silla de ruedas. Este modo debe optimizar los recursos para este colectivo.
- Permitir la modificación de los parámetros de retardo y amplitud de movimientos en tiempo real y ejecución de cada movimiento en un hilo diferente de forma independiente.
- Obtener una optimización de recursos con el fin de que el programa utilice los mínimos posibles.
- Eliminar el proceso de calibrado presentado en [1] con el fin de mejorar la experiencia de usuario.
- Permitir el guardado de sesiones y de datos personales de usuario, así como de configuraciones de interfaz de movimientos definidos.

- Permitir registrar y recoger en tiempo real datos de uso de cada sesión y de los movimientos empleados.
- Creación de una interfaz simple e intuitiva con navegación de ventanas para la ejecución de todas las funciones del programa y el registro de datos y sesiones.

1.3. Estructura

La memoria se estructura en un total de cinco capítulos y dos anexos. En el primer capítulo se proporciona una visión general del proyecto, así como las motivaciones que han llevado a su desarrollo y los objetivos específicos del mismo.

En el segundo capítulo se hace un análisis detallado de los antecedentes y la tecnología empleada en la realización del proyecto. Esta información es de interés para el lector para comprender los capítulos siguientes y tener una visión general de las herramientas existentes hasta el momento.

En el tercer capítulo se presenta la descripción del programa implementado prestando principal atención a los aspectos que afectan directamente al usuario y su experiencia con el programa, como es el desarrollo de la interfaz gráfica del programa y de la elección de los modos especializados para personas con discapacidad.

En el cuarto capítulo se proporcionan los resultados y su posterior análisis de las pruebas realizadas con el programa. En ellas han participado dos asociaciones de personas con discapacidad y voluntarios con el fin de recoger datos que ayuden a evaluar el uso real del programa y su compatibilidad con otras aplicaciones.

En el quinto y último capítulo se proporcionan las conclusiones a las que se ha llegado tras el desarrollo del programa y la realización de las posteriores pruebas con voluntarios, evaluando sus capacidades en casos reales. Finalmente se ofrece una visión general de las posibles mejoras a largo plazo que se pueden introducir en Mokey.

En el Anexo 1 se proporciona un presupuesto relativo a los costes materiales y horas de trabajo de la realización del proyecto y, finalmente, en el Anexo 2 se proporciona un manual de usuario para el uso del programa.

Capítulo 2. ANTECEDENTES

2.1. La discapacidad como problema social

En este apartado se busca informar acerca de todos los aspectos y líneas que afectan a la realización del trabajo en el ámbito social y tecnológico. Antes de realizar un proyecto, es necesario informarse sobre las líneas de estudio que actualmente están abiertas, las investigaciones previas que se han realizado y el marco actual de la situación que se ve afectada en el trabajo durante el proyecto.

En la actualidad, cerca de 1000 millones de personas viven con algún tipo de discapacidad [4]. El mundo de una persona con discapacidad puede ser muy diferente en función del tipo de invalidez, del lugar donde se resida o de la capacidad de acceso a recursos que tenga cada persona. A medida que pasa el tiempo, la población envejece más, provocando un crecimiento en el número de casos y en el número de tipos de discapacidad. Incluso, es altamente probable que la mayoría de las personas sanas padezcan algún tipo de incapacidad en algún momento. Por ello, se busca facilitar el acceso a recursos y herramientas que ayuden a estas personas en su día a día de forma que puedan realizar cualquier tarea independientemente de su discapacidad. Cerca de la cuarta parte de las personas que tienen alguna discapacidad no tienen acceso a los recursos necesarios para vivir como una persona más, siendo dependiente de otras personas o instituciones. Un recurso muy importante para luchar contra esta barrera es la tecnología.

Gracias a la tecnología se pueden implementar sistemas y herramientas que ayuden en la vida diaria de estas personas que por alguna razón tienen impedimentos para realizar tareas y trabajos cotidianos con total normalidad. La discapacidad no debe ser un obstáculo para el éxito, sino una puerta que se puede atravesar con la herramienta adecuada.

En la actualidad, las personas con discapacidad tienen un índice mayor de pobreza, de enfermedades, menos esperanza de vida y peor tasa de resultados académicos. Por ello se necesitan herramientas que minimicen este impacto y acerquen estos índices y tasas a los de la población sin discapacidad alguna. Es importante que cualquier persona con discapacidad pueda optar a ser el mejor físico del mundo o dedicarse a la profesión que más le gusta sin ningún impedimento físico, psicológico ni discriminatorio.

Según la Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud (CIF), la discapacidad se define como “término genérico que engloba deficiencias, limitaciones de actividad y restricciones para la participación” [4]. Existen muchos tipos de discapacidades y todas ellas se dividen en dos grandes grupos:

- **Discapacidades físicas:** Problemas en el sistema motor, sistema nervioso o deficiencias en la estructura ósea o de tipo muscular entre otras.
- **Discapacidades psíquicas del tipo intelectual o sensorial:** problemas en la interacción con otras personas, enfermedades del tipo Síndrome de Down, Síndrome de Asperger, depresión, falta de atención, etc.

2.2. La tecnología como posible solución

La principal razón de ser de la tecnología es la de facilitar la vida a las personas, a la población. En la actualidad existen multitud de herramientas y procedimientos tecnológicos que son empleados diariamente por muchas personas, optimizando el tiempo, las tareas y permitiendo poder hacer acciones que sin la tecnología nunca hubiera sido posible. Un ejemplo de ello es la

informática, con la que han venido los sistemas y herramientas que permiten la realización de gran parte de las acciones que se realizan.

En el ámbito de la discapacidad, la tecnología ayuda a abordar cada uno de los problemas de los diferentes tipos de discapacidades. En la actualidad se ha implementado, gracias a la tecnología, muchos de los productos diseñados para abordar los problemas a los que se enfrenta cualquier persona con discapacidad:

- **Problemas en la atención sanitaria:** Los principales problemas radican en la dificultad de comunicación, de aportar información e incluso obstáculos físicos. Todo ello impide el disfrute del sistema sanitario en igualdad de condiciones.
- **Problemas en la rehabilitación:** Es importante dar acceso a la rehabilitación de toda persona con discapacidad ya que este proceso busca generar capacidad humana y ayuda a una mejor inserción en la sociedad.
- **Problemas en la asistencia y en los servicios de apoyo:** Facilitar el acceso a los servicios de ayuda y soporte a discapacitados es fundamental para promover la autonomía y participación en la sociedad de aquellas personas con algún problema de discapacidad.
- **Gran obstáculo de la educación y en el empleo:** Este es uno de los problemas que más afecta y fomenta el crecimiento de las discapacidades. Existe un gran impedimento en la inserción en los ámbitos educativos y laborales de cierto grupo de personas, que, además, impide que vivan con normalidad. Se necesita una educación universal para acabar con este y cualquier tipo de discriminación ya sea por aspecto físico o psíquico.
- **Problema en el entorno:** Es el principal impedimento de toda persona con discapacidad para realizar una vida plenamente normal. En el entorno social es necesario que existan productos para paliar estos problemas en el ocio, el transporte, comunicación y luchar contra el aislamiento, la dependencia y la inserción social de este grupo de personas.

En todos estos problemas, la tecnología cobra un papel muy importante como medio para facilitar esta inserción social. A día de hoy existen multitud de herramientas fabricadas gracias a la tecnología que facilitan estos aspectos y permiten el acceso de cualquier persona, tenga o no algún tipo de incapacidad, a cualquier sistema social de ocio, entretenimiento, sanitario, educacional, etc.

En el ámbito del ocio y el entretenimiento, hay mucho desarrollo de sistemas de accesibilidad que permiten a las personas con discapacidad realizar acciones que no podrían realizar sin estos sistemas. Un ejemplo de ello son los programas de ordenador que permiten la utilización del mismo a aquellas personas que por problemas físicos o psíquicos no podrían manejarlo como cualquier persona sana. Muchos de estos programas y configuraciones vienen instalados en los propios sistemas operativos (SO) y pueden ser utilizados nada más encender el ordenador o configurarlo por primera vez.

Microsoft Corporation, uno de las empresas más importantes en desarrollo de software, proporciona herramientas en sus sistemas operativos y programas de productividad que se adecúan a diferentes tipos de discapacidad: visual, auditiva, habla, movilidad, edad o mentales [6]. El SO Windows 10 incluye herramientas como temas de alto contraste, lupa, narrador,

traductor por voz, retraso en notificaciones, subtítulos, teclado por voz o en pantalla, filtro de teclado, modo de lectura, etc. Además, cabe la posibilidad de instalar software de terceros para adaptar cada equipo a la necesidad de cada persona.

En la parcela de los videojuegos, hay muchas personas con discapacidad que, por su propia condición, no pueden jugar con normalidad a los videojuegos del mercado. En la actualidad existen herramientas que prestan soporte a estas personas y permiten que puedan utilizar cualquier tipo de videojuegos o programas. Un ejemplo de ello son programas como KinEmote, WIN+i [7] o FFAST (*Flexible Action and Articulated Skeleton Toolkit*) [8] [9], este último se explica con detalle en el apartado 2.4:

- **KinEmote:** Proyecto desarrollado por la empresa Evoluce en el año 2011 hasta el año 2013, cuando se abandonó su desarrollo para continuar la especialización de otros programas desarrollados por la compañía como Win+i. Esta Interfaz Natural permite controlar otras aplicaciones del ordenador por medio de la simulación del movimiento del cursor del ratón además de permitir simular la pulsación del botón izquierdo del mismo y ocho teclas del teclado. Surgió como desarrollo para control de televisiones Samsung.
- **Win+i:** Aplicación también diseñada por la empresa Evoluce en el año 2012 y que se continúa comercializando en 2017. Consiste en una Interfaz Natural desarrollada para simular el movimiento de ratón empleando el kit de desarrollo de la fundación OpenNI. En este caso mejora la precisión, estabilidad, respuesta y versatilidad de KinEmote. Este programa introduce la simulación de todas las teclas del ratón mediante movimientos de la misma y no mediante gestos. Se vende junto con otras cuatro aplicaciones especializadas en un aspecto diferente: *Image and Video Viewer for Kinect for Windows*, *PowerPoint Presenter for Kinect for Windows*, *Software for Kinect for Xbox* y *Media Viewer for Kinect for Windows* [7].

Todos estos programas son Interfaces Naturales ambientadas y creadas para facilitar la relación del ser humano con la máquina u ordenador y pueden ser utilizadas por personas con discapacidad para poder realizar multitud de tareas. Cada uno de estos programas está diseñado para el control de Microsoft Windows y distintas aplicaciones instaladas en el sistema mediante el dispositivo Microsoft Kinect, con la ayuda de sonidos, imágenes y gestos.

2.3. Microsoft Kinect

En el ámbito de la Realidad Aumentada, Microsoft desarrolla el dispositivo Microsoft Kinect. Gracias a este dispositivo, es posible recoger información referente a movimientos y gestos de personas y de esta forma ofrecer nuevas posibilidades en sectores como en el de la rehabilitación y la integración de personas con discapacidad.

La primera versión de Kinect, también conocida como Kinect 360 para Xbox, es una cámara basada en imágenes captadas por diferente tipo de sensores [5]. Fue lanzada al mercado en el año 2010 de la mano de Microsoft para competir con la cámara que existía por entonces, la Prime Sense. Esta cámara se diseñó para mejorar la experiencia de los jugadores de la por entonces novedosa Xbox 360. En los inicios se lanzó una sola versión exclusiva de la videoconsola Xbox 360, aunque también compatible para ordenador con Microsoft Windows 7 y Microsoft Windows 8 mediante un adaptador USB. Kinect compite con Wiimote y PlayStation Move en el ámbito de los videojuegos.

Kinect permite a los usuarios interactuar de una forma alternativa a la hora de jugar con la videoconsola Xbox ejecutando las acciones que cada juego requiera sin necesidad de ningún contacto con mandos y controles, sino con movimientos y gestos corporales. Este dispositivo facilita interactuar y entender la forma de jugar como un concepto diferente revolucionando en ese momento el mercado del videojuego. La cámara Microsoft Kinect y su SDK actúan como Interfaz Natural de usuario reconociendo gestos, movimiento corporal y facial y es capaz de reconocer sonidos, comandos de voz u objetos. En [10] se muestran sus principales características y especificaciones técnicas.

La Kinect tiene mucho potencial en sistemas orientados a la salud y rehabilitación de las personas. En [11], [12], [13] y [14] se muestra la capacidad de este dispositivo para luchar contra las barreras que presenta la discapacidad para muchas personas y se sientan las bases para los posibles usos que Kinect puede desempeñar en el ámbito de la salud. Este dispositivo crea un precedente para acercar la Realidad Virtual al sector.

Para su utilización en ordenadores, en los inicios de la versión de Kinect en 2010, esta estaba limitada a unas funcionalidades básicas de un paquete instalable de Microsoft. A finales de ese mismo año, la fundación OpenNI [15] trabajó para publicar una SDK y unos drivers de código abierto para poder desarrollar aplicaciones independientemente del programa proporcionado por Microsoft y aumentando las posibilidades que ofrecía por entonces Kinect. Visto este potencial, poco tardó Microsoft en lanzar una SDK propia y un conjunto de *drivers* y herramientas de desarrollo para abrir todavía más el abanico de posibilidades que podía ofrecer esa cámara [16]. A principios de 2012, Microsoft lanza al mercado una Microsoft Kinect especialmente diseñada para ordenadores, la denominada *Kinect for Windows* [17]. Las principales diferencias entre la primera versión de Kinect y la versión especializada para Windows se detallan en la tabla 2.1.

Tabla 2.1: Principales diferencias entre Kinect 1.0 y Kinect for Windows.

	Kinect 1.0 for Xbox 360	Kinect for Windows
Año	2010	2012
Audio	Audio de 16 bits a 16 kHz	Audio de 32 y 64 bits
Cámara RGB	640 x 480 (32 bits) a 30fps	640 x 480 (32 bits) a 30fps 640 x 480 a 15fps RAW YUV 1280 x 960 RGB a 12fps
Sensores profundidad	320 x 240 (16 bits) a 30fps	80 x 60 (16 bits) a 30fps 320 x 240 (16 bits) a 30fps 640 x 480 (16 bits) a 30fps

La versión de Kinect orientada al ordenador presentó una serie de mejoras en los sensores y en la rapidez y precisión del procesado de datos además de introducir nuevas posibilidades de configuración de resolución y formato de captura. Además, introdujo una mejora muy importante destinada al uso al que se le iba dar, un “modo cercano”. Este nuevo modo permitía el uso de Kinect para reconocimiento facial o de alguna extremidad en concreto con mayor detalle de puntos permitiendo el reconocimiento de los mismos a distancias de 40 centímetros. En la figura 2.1 se puede observar la detección de los puntos en “modo cercano” de la mano izquierda. Este nuevo modo, sin embargo, no tiene un funcionamiento preciso, para según qué uso, en cuanto a detección de puntos se refiere, por lo que su uso no ha sido muy extendido.

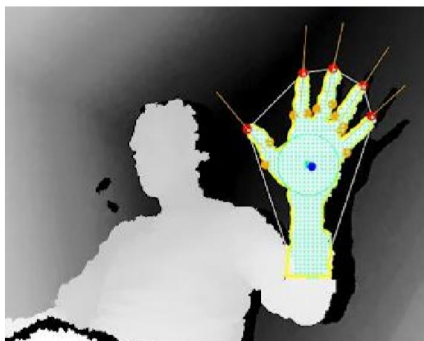


Figura 2.1: Detección de los puntos en “modo cercano” mediante Kinect for Windows. [18]

Ambas versiones de Kinect, tanto la primera diseñada para Xbox 360 como la diseñada para ordenador, son completamente compatibles con el sistema Windows. Para la primera, es necesario adquirir el adaptador específico. Ambas necesitan que los drivers y el kit de desarrollo SDK correspondiente estén instalados, ya sea el propietario de Microsoft o la librería de OpenNI. Físicamente ambas versiones presentan una apariencia similar sin alteración ninguna pese a distar cerca de dos años de diferencia en cuanto a fechas de lanzamiento al mercado. Esta mejora para ordenador introdujo la novedad de poder instalar varias Kinect en el mismo ordenador, algo imposible hasta ese momento. En cuanto al audio, la denominada *Kinect for Windows* introdujo una mejora en la detección de audio y en el número de idiomas soportados aumentando de forma considerable su número.

La cámara Kinect consta de una estructura compleja como se puede apreciar en la figura 2.2. En ella se observa que la Kinect está formada por un sensor RGB (Componentes *Red*, *Green* y *Blue*), dos sensores infrarrojos de profundidad, un *array* de micrófonos y un motor destinado a la rotación del pie de la misma. La resolución y distintas posibilidades de configuración de cada sensor según la versión se detallan en la tabla 2.1.

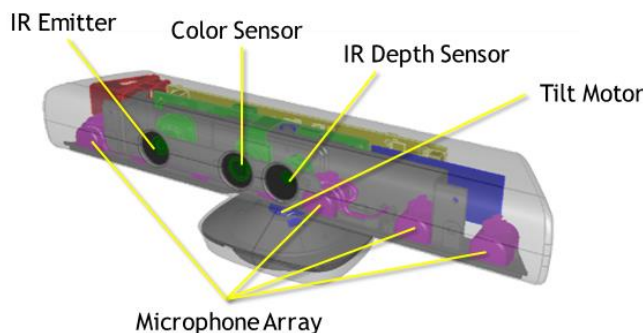


Figura 2.2: Estructura de los distintos sensores de Kinect 1.0 y Kinect for Windows. [17]

En primer lugar, el sensor RGB actúa como un cámara convencional captando una imagen de color en las tres componentes básicas de color: *Red*, *Green* and *Blue*. Los dos sensores de profundidad son dos sensores CMOS de tipo infrarrojo los cuales actúan, uno como emisor y otro como receptor de información para formar entre los dos una imagen de profundidad. En cuanto al audio, la Kinect cuenta con un *array* de pequeños micrófonos a cada lado de los sensores encargados de recibir información del audio captado. Además, la cámara está dotada de un motor para gestionar la inclinación en tiempo real en el plano vertical de la misma para su adaptación según la necesidad en cada momento.

El funcionamiento de todos los sensores en conjunto se combina de forma que los rayos infrarrojos parten del sensor infrarrojo emisor y, tras rebotar en las superficies expuestas delante del sensor, vuelve al sensor infrarrojo receptor para recoger la información de tiempo de cada uno de los rayos infrarrojos emitidos. Con esta información de tiempo de cada rayo, el sistema puede deducir la distancia debido a su longitud de onda de rayo definida y de esta forma definir una imagen de profundidad. En la figura 2.3 se pueden observar las dos imágenes captadas por la cámara, la imagen RGB y la imagen de profundidad resultado de la información captada por los dos sensores infrarrojos. Con todo ello, Kinect es capaz de reconocer a dos personas de forma simultánea y 20 puntos en el espacio pertenecientes a las articulaciones y puntos más importantes de cada una de las dos personas, siempre que se encuentren entre 1,2 y 3,5 metros de distancia de la cámara. Si capta más personas es posible que interfieran datos erróneos.

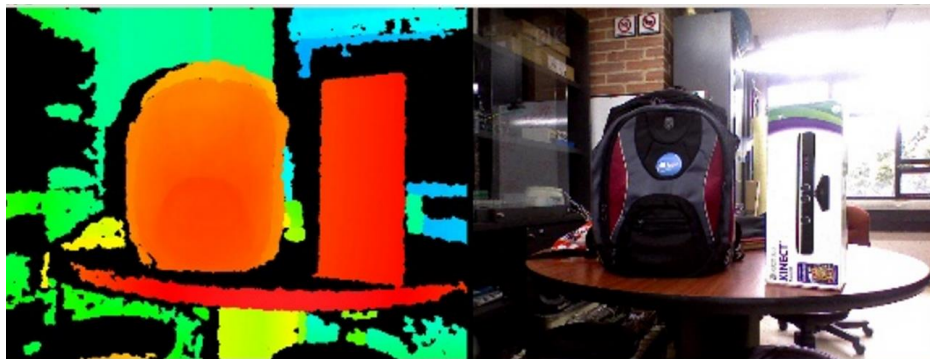


Figura 2.3: Imagen de profundidad (izq.) e imagen RGB (der.) captada de Kinect. [18]

Como ya se ha comentado, Kinect es capaz de captar 20 puntos diferentes en cada persona que se encuentre frente al sensor hasta un máximo de 2 personas, es decir; 40 puntos diferentes de forma simultánea. De cada punto es posible obtener, mediante la SDK, las tres coordenadas cartesianas X, Y y Z. En la siguiente figura 2.4 se observan cada uno de estos 20 puntos.

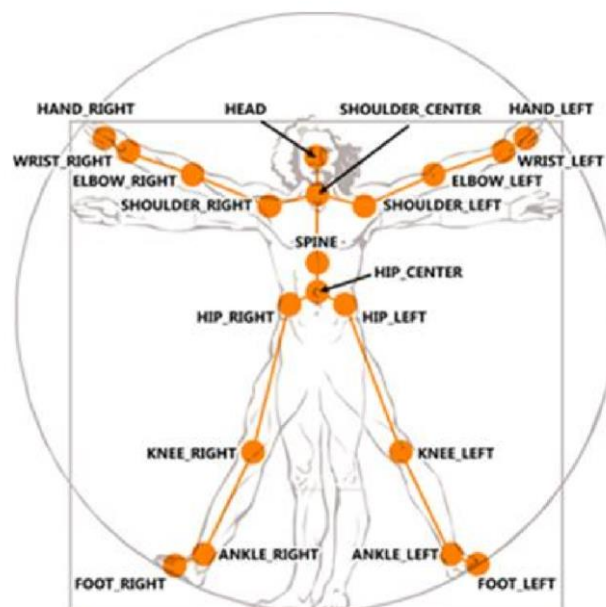


Figura 2.4: Representación los puntos del esqueleto humano que capta Kinect simultáneamente.

En 2013, Microsoft lanza una actualización de la cámara sacando al mercado la denominada Microsoft Kinect 2.0 [19]. Este sensor acompañó la campaña publicitaria de la por entonces nueva Xbox One. Esta cámara introdujo la posibilidad de captar imágenes con resolución fullHD (1080i) y la capacidad de captar a 6 personas de forma simultánea. En cuanto al aspecto, Microsoft sí que incluye un novedoso nuevo diseño. En este caso no se ha presentado ninguna versión a día de hoy especializada en ordenador, permitiendo su uso tanto en Xbox One como en ordenador con el mismo dispositivo, tan solo es necesario el adaptador correspondiente.

A día de hoy, la primera versión de Kinect, o en su defecto la versión de *Kinect for Windows*, ha mantenido un mayor soporte y desarrollo por parte de la comunidad frente a la más reciente Kinect 2.0. Esto es debido a la flexibilidad del kit de desarrollo correspondiente a cada dispositivo y las posibilidades que estos kits ofrecen al desarrollador.

2.4. FAAST

Flexible Action and Articulated Skeleton Toolkit, también conocida como FAAST, es una Interfaz Natural que permite simular eventos del teclado mediante movimientos corporales por medio de Microsoft Kinect. Con ello, facilita la accesibilidad a programas y características hasta ese momento complicadas para personas con discapacidad: permite controlar aplicaciones del ordenador mediante movimientos. El proyecto guarda relación con los objetivos que presenta Mokey, pero en este caso, se busca una mayor facilidad en la configuración, ampliación y optimización de funciones y mejora en el rendimiento.

Tiene la posibilidad de poder gestionar la información proporcionada por el dispositivo Kinect con la ayuda de los drivers y el kit de desarrollo, tanto de OpenNI como los drivers propietarios de Microsoft. Este hecho dota a la aplicación de una mayor adaptabilidad y capacidad de uso en un mayor número de ordenadores debido a que es posible elegir entre cualquiera de las dos librerías, incluso, permite tener ambas instaladas de forma simultánea. Además, permite la simulación del movimiento del cursor del ratón por medio del movimiento corporal dotando al programa de mayor posibilidad de juego y compatibilidad en multitud de aplicaciones.

Este proyecto nace en el año 2011 de mano de South California University hasta evolucionar a su última versión 1.2 a finales del año 2013 [9]. Esta Interfaz Natural permite una configuración personalizable, versátil y adaptada a cada situación. Es posible la configuración de todos los movimientos que se requieran, su precisión, rango de movimiento, rotación de articulaciones y la tecla del teclado físico asociada al mismo.

La estructura de la aplicación es de tres módulos diferenciados como se puede observar en la figura 2.5. El primer módulo es el dedicado a la configuración de la librería de desarrollo que se emplea en el procesamiento y toma de datos, configuraciones básicas del programa y del uso del mismo y, además, permite, en la pestaña *gestures*, poder configurar tantos movimientos como se requieran según las necesidades del usuario. En un segundo módulo, FAAST permite mostrar la imagen que capta Kinect en cada momento pudiéndose configurar qué información se quiere mostrar en esta ventana. En un tercer módulo, parte inferior de la figura 2.5, el programa muestra todos los registros, errores y estados de cada acción que realiza.

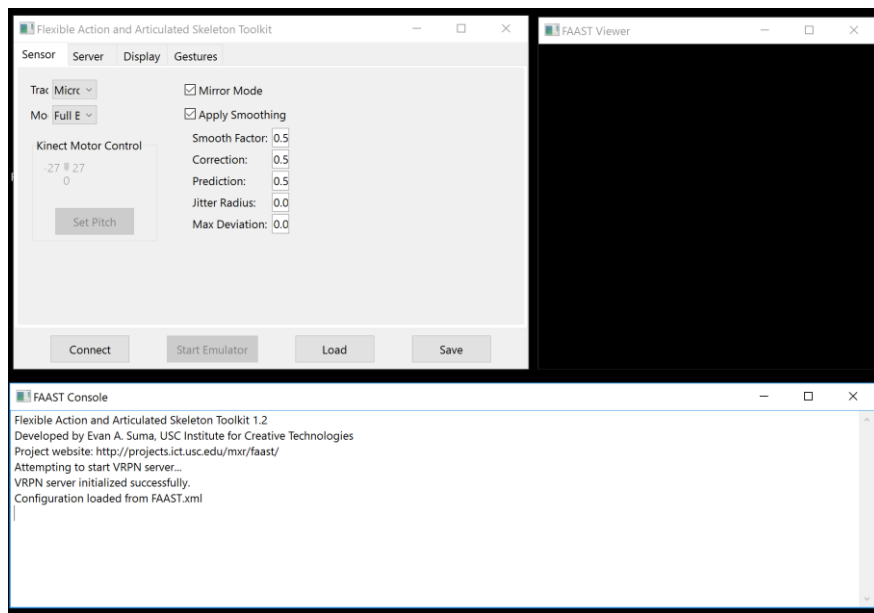


Figura 2.5: Apariencia principal de los tres módulos de FAAST

La principal característica que presenta FAAST frente al Win+i o KinEmote es la posibilidad de configurar multitud de movimientos con una gran personalización. Cada movimiento es posible configurarlo independientemente de los demás eligiendo la acción que se requiere a la entrada, es decir, la acción la cual ejecuta el evento que se quiere llevar a cabo, y por otro lado la acción de respuesta asignada al mismo. En la figura 2.6 se muestran las diferentes opciones que permite la configuración de cada movimiento. Una vez creado con su nombre correspondiente, mediante la interfaz es posible asignar diferentes acciones tanto en la entrada del movimiento como en la salida. En la parte inferior izquierda de esta figura se muestran los posibles eventos que se pueden asignar a la entrada del movimiento, mientras que a la derecha se muestran las acciones de respuesta posibles a la salida. Como entrada acepta valores fijos de tiempo, velocidad, rotación, posición de puntos del esqueleto o parte del cuerpo y como salida puede ejecutar eventos del teclado o del ratón además de otros eventos propios del programa FAAST o de configuración de Kinect. Es una Interfaz Natural altamente configurable.

El hecho de que sea un programa altamente configurable también juega en contra de la usabilidad del mismo debido al alto grado de complejidad para el usuario medio. Al tratarse de una configuración en la que el usuario tiene que determinar de forma manual todos los datos importantes de cada acción o movimiento configurado, utilizar esta Interfaz Natural conlleva tener un conocimiento previo, tanto de las articulaciones que es capaz de detectar Kinect, como de la propia configuración de tiempos, velocidades y rangos de cada acción configurada. FAAST permite configurar un alto número de movimientos sin límite alguno de si son posibles o realizables, por lo que es necesario ese conocimiento y control de los aspectos básicos de Kinect.

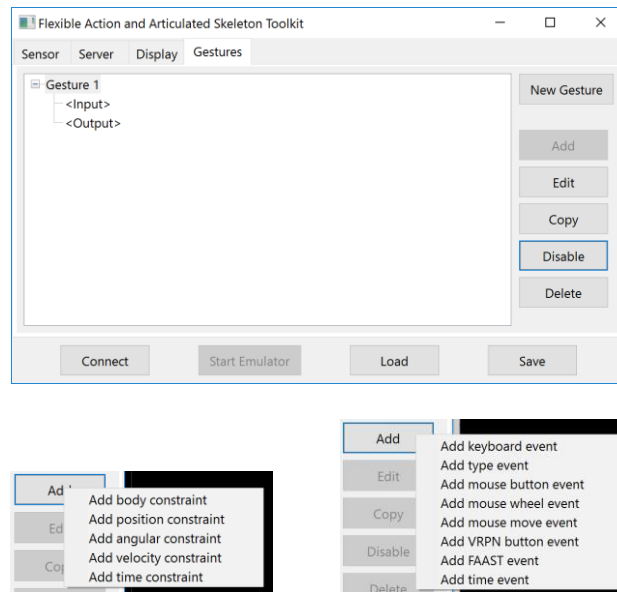


Figura 2.6: Ventana de configuración de movimientos en FAAST (arriba), eventos seleccionables para la entrada de cada movimiento (abajo izquierda) y eventos seleccionados para la salida de cada movimiento (abajo derecha).

2.5. Otras tecnologías

Además de FAAST, otras organizaciones han desarrollado soluciones similares, pero sin embargo, no tan completas. Se han desarrollado programas para paliar los problemas de compatibilidad entre las Interfaces Naturales y las aplicaciones de Windows como el procedimiento descrito en [20]. De esta forma se permite establecer una selección de aplicaciones en las que actúa la Interfaz Natural. En [21] se establecen las bases para el desarrollo de Interfaces para el control de eventos de teclado y ratón para de esta forma ampliar el rango de posibilidades que se pueden ofrecer a personas con discapacidad y que sientan las bases para el desarrollo de aplicaciones como FAAST o Mokey, de la que trata este texto.

Para permitir el desarrollo de Interfaces Naturales, es necesario un conjunto de herramientas de desarrollo que permitan interactuar entre los datos proporcionados por los *drivers* de Kinect y la aplicación que se quiere controlar. En los inicios de Kinect, surgen multitud de estudios sobre el seguimiento del movimiento corporal mediante estas herramientas como se puede leer en [22], [23], [24] y [25]. La mayoría de estos estudios se han realizado sobre la librería OpenNI, pero sin embargo es aplicable a la más reciente SDK de Microsoft Kinect. El lanzamiento de este kit de desarrollo de forma oficial trae consigo el abandono de la librería de OpenNI y con ello los proyectos derivados de la misma, proyectos más ambiciosos y con más posibilidades como es el caso de FAAST.

Tras multitud de estudios relacionados con el seguimiento y control del movimiento corporal, surgen diferentes líneas de desarrollo relacionados con el reconocimiento de los gestos de la mano. Esto presenta un punto de inflexión en el desarrollo de este tipo de aplicaciones ya que se abre el abanico de posibilidades de las Interfaces Naturales para Kinect, como en reconocimiento de lenguas de signos. Sin embargo, no se puede olvidar la dificultad en la precisión de los datos de Kinect en estos gestos. En [26] se muestra un ejemplo del estudio y desarrollo de estas tecnologías orientadas a gestos de las manos y los brazos.

Capítulo 3. DESCRIPCIÓN TÉCNICA DE MOKEY

A lo largo de este tercer capítulo, se va a exponer con detalle los aspectos desarrollados que afectan al usuario, ya sea a la hora de utilizar la Interfaz Natural Mokey como a la hora de configurar su uso. Se presta principal atención a funciones del usuario como son la configuración de cada uno de los movimientos y los parámetros que es posible personalizar dando una explicación del por qué se han elegido esos parámetros y el cómo se han de emplear para un uso óptimo. En [1] se detalla la implementación de estos conceptos de configuración atendiendo a los algoritmos, librerías y funciones empleadas. A lo largo del capítulo se referencia este documento con objeto de facilitar la comprensión del lector y poder tener una visión global de Mokey en todos los aspectos.

3.1. Software y Hardware empleados

En el primer apartado de este capítulo, se detallan las herramientas empleadas en el desarrollo del proyecto, así como la descripción de cada una de ellas atendiendo a las características y versiones empleadas.

3.1.1. SDK v1.8 (*Software Development Kit*)

En primer lugar, es necesario obtener el conjunto de herramientas, librerías y funciones necesarias para interactuar con la cámara Kinect y poder interpretar los datos de la misma para procesar la información necesaria y elaborar el código fuente de la aplicación. Este conjunto de herramientas, junto con los *drivers* del Sistema Operativo, forman el kit de desarrollo software o en su defecto SDK (*Software Development Kit*). Además, este SDK incluye aplicaciones básicas que ayudan a comprender el funcionamiento y funciones principales de Kinect.

La versión del SDK empleada para la realización del proyecto es la versión 1.8 lanzada por Microsoft en septiembre de 2013 [16]. La elección de este SDK en detrimento del de OpenNI es debido al alto desarrollo de la nueva solución de Microsoft y al alto respaldo de la comunidad de desarrolladores, lo que provocó el cese del desarrollo de la solución de la fundación OpenNI. Esta versión 1.8 es la última, considerada definitiva, del conjunto de herramientas de desarrollo destinado a la primera Kinect en sus dos versiones, la de Kinect de Xbox360 y la de *Kinect for Windows*. Esta versión, además, es compatible con todas las aplicaciones desarrolladas con funciones y librerías de las antiguas versiones de SDK al igual que con los sistemas operativos Windows 7, 8 u 8.1, ya sea de 32 o de 64 bits.

Este kit de desarrollo permite realizar aplicaciones mediante dos lenguajes de programación diferentes: C# [27] y C++. Incluye las funciones necesarias para poder interpretar los datos captados con Kinect y, mediante el diseño de algoritmos, conseguir aplicaciones y programas con sus funciones concretas.

Entre las funciones más destacadas de este kit de herramientas de desarrollo se encuentran aquellas destinadas al seguimiento del esqueleto captado por Kinect, así como de la gestión de todas las coordenadas y puntos calculados en cada instante tras la detección de imágenes del dispositivo. Con estas funciones, es posible dibujar el esqueleto captado en el ordenador y recoger toda la información del mismo para programar más adelante las funciones principales de Mokey. Estas funciones de seguimiento corporal son las más importantes del kit de desarrollo, pero además de estas, este kit contiene funciones destinadas al seguimiento facial, para poder detectar y gestionar gestos y datos del rostro, funciones destinadas a la detección por voz o funciones propias para gestionar Kinect y el motor de la base de la misma. El kit de desarrollo también incluye funciones relativas al procesamiento de la información recogida y

funciones destinadas a la edición del esqueleto. Por ejemplo, para poder eliminar partes del esqueleto o ser intercambiadas por otras de un esqueleto diferente.

3.1.2. Lenguaje C#

Uno de los lenguajes admitidos y el elegido para la realización del proyecto es C#. Esta elección es debido a su alta compatibilidad con el SDK de Kinect y con el software Visual Studio. C# es un lenguaje orientado a objetos y nace de la mano de Microsoft en el año 2000 aunque no fue estandarizado hasta el año 2006 bajo la norma ISO/IEC 23270 [28]. Este lenguaje nace para hacer frente al tan popular Java y está basado en otros lenguajes como C o C++. C# es un lenguaje innovador y que introduce un nuevo concepto a la hora de realizar interfaces gráficas en los programas debido a la cantidad de recursos que este ofrece para estos propósitos, como un editor de interfaces gráficas propio y un depurador integrado. Permite a los desarrolladores implementar y compilar soluciones seguras que se ejecutan bajo la arquitectura .NET Framework [27].

3.1.3. Microsoft Visual Studio 2013

Para el desarrollo de Mokey es necesario un entorno de desarrollo mediante el cual, por medio del lenguaje de programación C# y las funciones del SDK de Kinect, se desarrollen los algoritmos y funciones necesarias para desarrollar y compilar el código fuente del programa. Microsoft proporciona diferentes versiones de Visual Studio 2013 para adaptar cada una a las necesidades del programador. El entorno de desarrollo que mejor se ajusta a las necesidades del kit de desarrollo de Kinect y el lenguaje C# con el que se trabaja es el programa Microsoft Visual Studio Ultimate, en su versión del año 2013 [29].

Microsoft Visual Studio 2013 es un entorno de desarrollo creado por Microsoft que permite, mediante diferentes lenguajes, la edición, depuración y compilación de aplicaciones basadas en entorno .NET Framework para Windows. Contiene un conjunto de herramientas y tecnologías de desarrollo software para poder crear aplicaciones de alto rendimiento y gran efectividad. .NET Framework proporciona un entorno de programación coherente para poder crear dichas aplicaciones. Este entorno de desarrollo se caracteriza por su herramienta de edición de interfaces gráficas, muy visual y con la que sin un conocimiento avanzado es posible crear interfaces de usuario simples con las que poder interactuar con el programa en cuestión.

Con Visual Studio es posible generar muchos tipos de proyectos bajo diferentes plantillas y bajo diferentes lenguajes de programación, como se observa en la figura 3.1. En la figura 3.2 se observa el espacio de trabajo para edición de interfaces gráficas, el punto fuerte del programa. En la parte izquierda se observan todos los elementos que es posible utilizar en la interfaz mientras que en la parte central superior se muestra el espacio de edición de la interfaz y en la parte central inferior el código fuente que hace referencia a la misma. En la parte derecha es posible editar los parámetros de cada uno de los elementos de la interfaz además de los eventos que los relacionan con la parte del código de programa de la aplicación en cuestión.

Visual Studio diferencia en dos archivos diferentes el código fuente de la interfaz gráfica, bajo un archivo *xaml*, y el código relativo a los algoritmos y funciones de la aplicación o programa en sí mismo. La vía de relación entre estos dos archivos de código es la generación de los denominados eventos en cada elemento de la interfaz para relacionar diferentes acciones de los mismos con funciones en el código del programa principal.

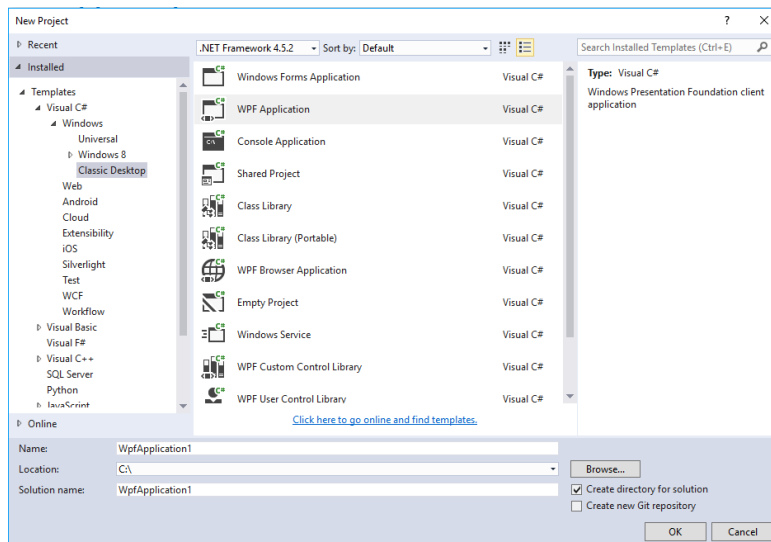


Figura 3.1: Asistente para la creación de proyectos dentro de Visual Studio 2013

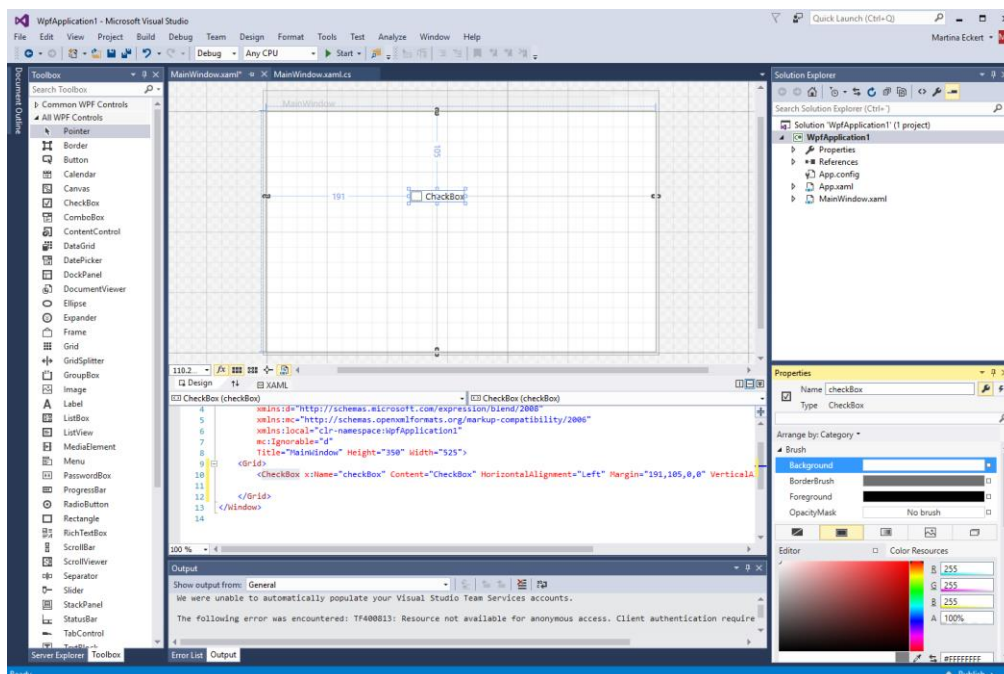


Figura 3.2: Editor de interfaz gráfica de un proyecto de Visual Studio 2013.

3.2. Función y descripción

Como ya se ha comentado anteriormente, Mokey es una aplicación diseñada para facilitar el uso de otras aplicaciones de ordenador a personas con discapacidad o con alguna insuficiencia motora. Por ello, Mokey nace bajo un concepto de simplicidad y optimización de recursos añadiendo un amplio rango de movimientos diferentes personalizables bajo dos parámetros principales: el retardo o *delay* y desplazamiento o *displacement*. Estos dos parámetros se detallan en los subapartados 3.2.1 y 3.2.2.

Con Mokey, es posible elegir entre doce distintos movimientos predefinidos y otros cuatro movimientos grabados. Todos estos movimientos se detallan en los apartados 3.4 y 3.5. Los

algoritmos y funciones que hacen posible estos movimientos se definen y explican en [1] así como también el funcionamiento de Mokey en cuanto a la gestión de todos los movimientos configurables. En [1] también se explica cómo cada uno de los movimientos se define en un hilo independiente creado a su vez por el hilo principal. Este hilo principal es el que sostiene la interfaz gráfica, la configuración de Mokey y el núcleo principal del proceso de los datos que capta Kinect. En la figura 3.3 se muestra un diagrama explicativo de cómo el hilo principal sostiene la propia interfaz gráfica y su configuración mediante la cual crea tantos hilos hijos como movimientos se activen en la misma. Cada uno de estos hilos hijos contiene sus propios valores de *delay* y *displacement* así como las variables relativas almacenadas, que se actualizan en tiempo real, de cada uno de los puntos del esqueleto con los que entra en juego cada movimiento configurado en Mokey.

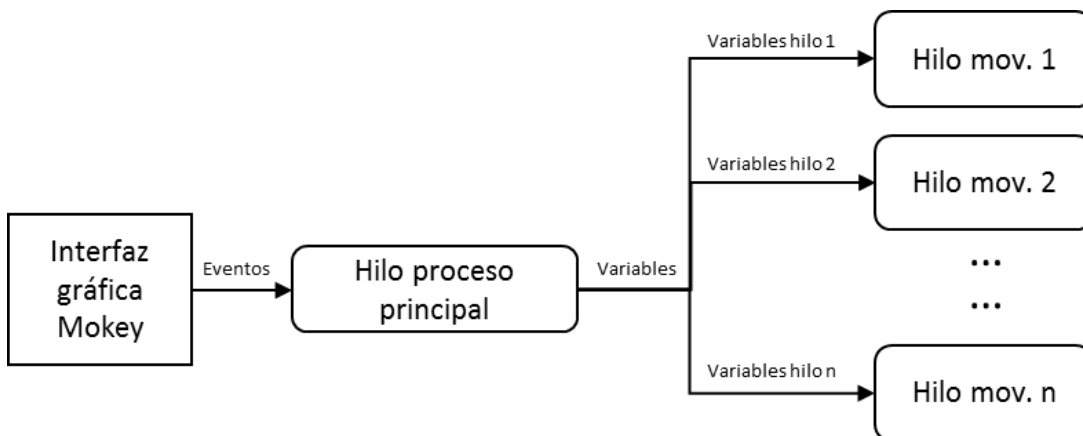


Figura 3.3: Diagrama de flujo entre interfaz, hilo principal e hilos secundarios.

En la figura 3.4 se observa un esquema con el flujo de trabajo de Mokey. En este esquema se muestran los algoritmos de detección de movimientos y de simulación de teclas diseñados para la aplicación y definidos detalladamente en [1]. En dicha figura se puede encontrar por separado los algoritmos que afectan al proceso principal, así como los algoritmos propios de cada hilo hijo relativo a cada movimiento.

El funcionamiento principal de Mokey se detalla en el flujo definido en la figura 3.4, donde con la información definida por el usuario en la interfaz y las coordenadas de los puntos del esqueleto que Kinect detecta de forma continua, se generan tantos hilos hijos como movimientos se configuren en la misma interfaz. Cada uno de estos hilos hijos evalúa las coordenadas de los puntos que afectan a cada movimiento comparando, según cada algoritmo, con el valor definido en el parámetro *displacement*. Si las condiciones de cada algoritmo definido en cada hilo hijo se cumplen, se ejecutan eventos de la tecla previamente configurada en la interfaz para tal movimiento con tanta frecuencia como se defina en el valor *delay*.

Dentro del hilo principal se establece el algoritmo más importante de Mokey, el que es capaz de obtener, mediante la información que se extrae de Kinect por medio de las funciones de su SDK, las coordenadas relativas de cada uno de los puntos del esqueleto. Estos datos se almacenan en variables locales del sistema para que cada hilo hijo evalúe si se debe simular el evento de teclado definido o no. En este bucle de registro de coordenadas también se realiza un registro de datos de los valores de las condiciones que evalúan si se realizan los movimientos o no. Este registro de datos se detalla en el apartado 3.6.

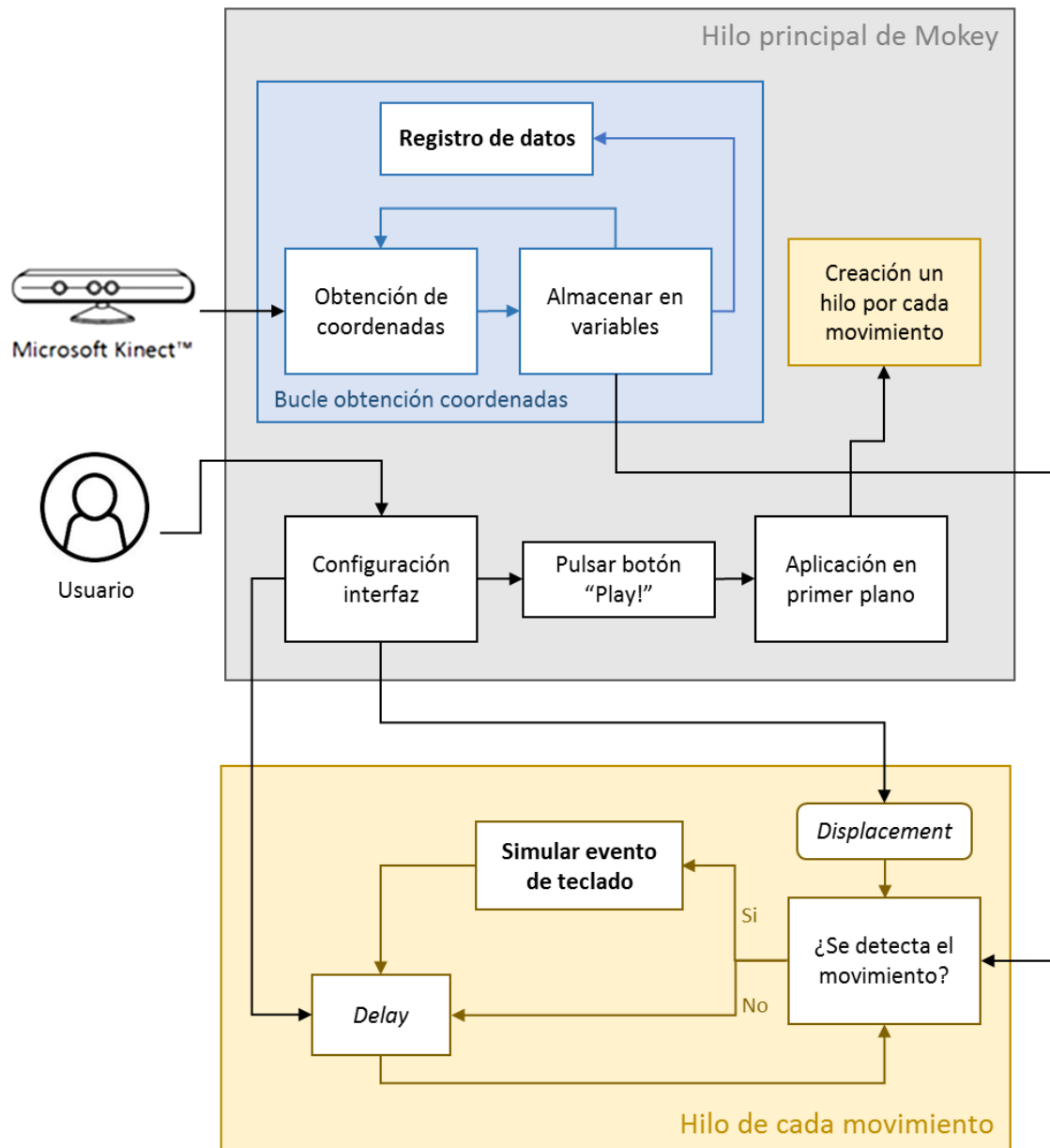


Figura 3.4: Diagrama del funcionamiento de Mokey

3.2.1. Concepto de *delay*

Como se puede observar en la figura 3.4, cada movimiento o hilo secundario de cada uno de los movimientos tiene entre sus parámetros individuales el valor de retardo o *delay*. Este retardo nace de la necesidad de establecer un control del número de veces que se quiere que se ejecuten eventos del teclado de forma consecutiva. Por ejemplo, para aplicaciones que requieran pulsar una tecla de forma repetida con mucha frecuencia, como correr en ciertos juegos, es necesario que ese retardo sea muy pequeño. Mokey está diseñado para proporcionar el manejo de aplicaciones de ordenador mediante gestos captados por Kinect. Muchas de estas aplicaciones requieren una única pulsación o pulsaciones de teclas con más periodo entre evento y evento. Un ejemplo de ello es el Tetris, ya que para la tecla de bajar la ficha es necesario un alto tiempo de retardo o de lo contrario bajarían varias a la vez. La descripción detallada de la implementación de los retardos se lleva a cabo en [1].

El objetivo de este retardo es el de conseguir que Mokey se adapte a las circunstancias de cada usuario y a las posibilidades de cada aplicación. Es posible que una persona con movilidad reducida necesite un retardo diferente entre eventos del teclado que otra persona sin discapacidad para poder disfrutar de una aplicación en igualdad de condiciones. Mokey busca poderse adaptar a las necesidades de cada persona sea cuales sean las condiciones de cada uno.

En Mokey es posible configurar el retardo con valores entre 0 y 600 milisegundos en cada uno de los doce movimientos predefinidos y en los cuatro grabados. Para comprender este concepto, si se selecciona para un movimiento determinado, un retardo de 500 ms, esto significa que el sistema ejecutará la función que establece los eventos de teclado cada medio segundo. Es decir, en un segundo el sistema simulará dos pulsaciones de teclado. Si se cumple la condición de dicho movimiento, o lo que es lo mismo, si se está realizando el movimiento definido de forma constante, estos eventos de teclado se producen con la frecuencia que se especifique en el valor retardo o *delay*. En la figura 3.5 se puede observar mediante qué botones se configura el retardo para cada uno de los movimientos. Más adelante, en el apartado 3.3, se detalla de qué objetos gráficos se componen.

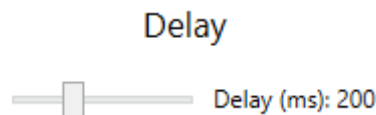


Figura 3.5: Representación de los botones de configuración del retardo en la interfaz gráfica.

3.2.2. Concepto de *displacement*

Al igual que el parámetro de retardo, cada movimiento configurado contiene su propio parámetro de sensibilidad, desplazamiento o *displacement*. Este parámetro hace referencia a la amplitud o desplazamiento relativo a cada movimiento configurado en Mokey, o lo que es lo mismo, cuánto hay que exagerar un movimiento para que Mokey lo interprete como realizado y ejecute el evento de teclado correspondiente.

Este parámetro no existe en los cuatro movimientos grabados debido a que se personaliza este aspecto a la hora de grabar el movimiento en cuestión. Mokey permite ajustar este parámetro entre 0 y 0.7 metros, aunque debido a la fisiología del cuerpo humano, no es factible para el cumplimiento de los movimientos configuraciones de más de 0.5 metros. Esto es debido a que la distancia entre los puntos del esqueleto que se compara en los algoritmos de cada movimiento siempre es menor. Sin embargo, configuraciones de este parámetro con valores pequeños, como 0.1 metros, permite que ese movimiento cumpla con frecuencia las condiciones de los algoritmos y se lancen con cierta facilidad los eventos de teclado.

En la figura 3.6 se muestra el aspecto de la configuración de este parámetro de desplazamiento, amplitud o *displacement* dentro de la interfaz gráfica de Mokey. En el apartado 3.3 se detalla más sobre la implementación de dicha interfaz gráfica.

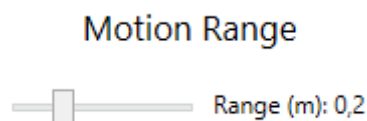


Figura 3.6: Representación de los botones de configuración del desplazamiento dentro de la interfaz.

3.3. Interfaz gráfica

En este apartado se establece una visión de la interfaz gráfica del programa Mokey y una descripción de cada una de las pantallas que tiene el mismo y de todos los elementos empleados en su programación.

La interfaz gráfica de Mokey está diseñada bajo una navegación fluida e intuitiva entre ventanas. El diseño de todas estas ventanas es simple, con objetos grandes y visibles de forma cómoda con fondo blanco y letras y objetos en negro. El objetivo de este diseño es el de evitar la fatiga visual y ser una interfaz lo más simple e intuitiva posible. En la figura 3.7 se muestra el diagrama referente a la navegación entre las ventanas de Mokey.

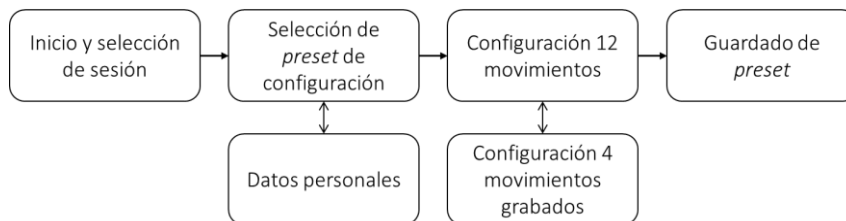


Figura 3.7: Diagrama navegación de ventanas de la interfaz gráfica de Mokey

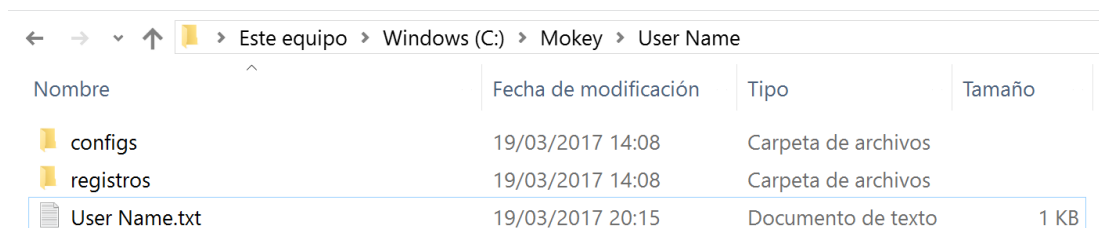
Inicio y selección de sesión

En primer lugar, al ejecutar Mokey, aparece una ventana inicial como la que se muestra en la figura 3.8. En esta ventana, se puede observar que se da la bienvenida al programa e invita a introducir un nombre. Este nombre es el relativo a la sesión de la persona que va a utilizar el programa, ya sea usuario nuevo o usuario con sesión previa abierta. En caso de ser sesión nueva, Mokey crea un nuevo directorio en la raíz del disco local dentro de la carpeta denominada “Mokey” con el nombre de la persona que inicia dicha sesión. Sin embargo, si el nombre se reconoce como de alguien que ya tiene una sesión abierta, Mokey carga todos los datos de sesiones previas, así como ficheros de configuraciones y de datos personales ya configurados anteriormente en otras sesiones. En este directorio de cada sesión se guardan los datos personales, configuraciones de sesiones previas y los registros que detallan más adelante en el apartado 3.6. En la figura 3.9 se observa cómo queda este directorio de cada sesión iniciada en Mokey. Todos los datos almacenados son exclusivos de la sesión en cuestión y, por ello, se almacenan en directorios independientes.



Figura 3.8: Pantalla de bienvenida y de inicio de sesión de Mokey.

Esta ventana de bienvenida se compone de un título de ventana descriptivo con posibilidad de cerrar y minimizar con el logo relativo a la letra “S” del logo del centro CITSEM, donde se ha realizado el proyecto. Además, también viene acompañada del logo en la parte central, un texto de bienvenida y un cuadro de texto donde se introduce el nombre de la sesión que se va a comenzar, por defecto “User Name”. Cuando este cuadro esté relleno es necesario hacer clic con el ratón en el botón “Start” para ir a la siguiente ventana: la configuración de los datos de la sesión. En la parte inferior de la ventana se nombran a los autores del proyecto y año de compilación de la versión de Mokey.



Nombre	Fecha de modificación	Tipo	Tamaño
configs	19/03/2017 14:08	Carpeta de archivos	
registros	19/03/2017 14:08	Carpeta de archivos	
User Name.txt	19/03/2017 20:15	Documento de texto	1 KB

Figura 3.9: Directorio de cada sesión en la raíz del disco dentro de la carpeta Mokey.

La introducción del concepto de creación de sesiones tiene el objetivo de que el programa pueda ser empleado como ayuda a especialistas para monitorizar datos de rehabilitación personalizados a cada paciente y realizar un seguimiento de cada usuario de Mokey. Cada sesión almacena datos personales y de configuración del programa de cada usuario que use Mokey de forma que dicho usuario pueda cargar y registrar sus propios datos personales tantas veces como quiera sin perder ninguna información.

Una vez ejecutado el botón “Start”, Mokey hace un cambio de ventana a la ventana de configuración de datos de sesión, como la que se muestra en la figura 3.10.

Selección de *preset* de configuración y datos personales

En esta nueva ventana de configuración de la sesión, Mokey muestra las configuraciones de la interfaz en cuanto a movimientos y parámetros ajustados que han sido guardados previamente en la sesión personal asociada al nombre introducido en la primera ventana. En la figura 3.10 se muestra una ventana con dos pestañas, una referente a *presets* de configuración ya guardados y otra referente a los datos personales, pestaña que se muestra en la figura 3.11.

En la primera pestaña con los *presets* o archivos de configuración se permite seleccionar configuraciones previas de Mokey. Este aspecto permite la posibilidad de guardar configuraciones específicas de aplicaciones o juegos y evitar su laboriosa configuración cada vez que se quiere utilizar una aplicación para un mismo fin. Esta pestaña está compuesta por un texto de orientación, acompañado de una lista de contenido que busca en el directorio C:/Mokey/UserName/configs todos los archivos relativos a *presets* que ya se encuentran guardados en la sesión actual. En la parte inferior se muestra un cuadro de texto con los comentarios asociados a cada uno de los *preset* de la lista acompañado del botón “Start” para hacer otro salto de ventana, en este caso a la ventana de configuración principal de Mokey.

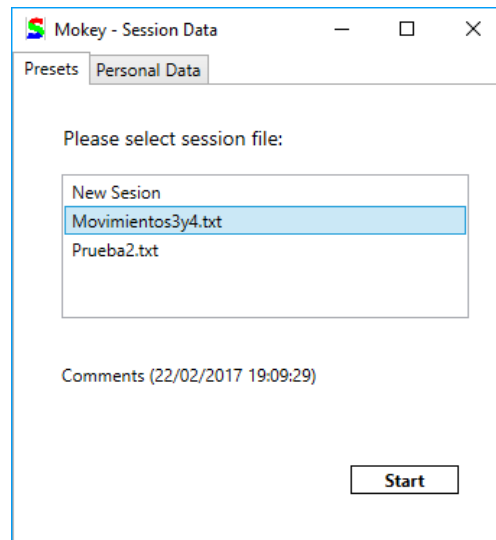


Figura 3.10: Pantalla de carga de configuración anterior dentro de la sesión personal de Mokey.

La segunda pestaña de esta ventana está destinada a los datos personales de la sesión del usuario actual, el que ha iniciado Mokey. Esta pestaña tiene la apariencia que se muestra en la figura 3.11 y muestra en diferentes campos de texto información personal de nombre, apellidos, edad, tipo de discapacidad y, por último, comentarios adicionales. Estos campos están destinados a poder extraer un informe detallado de cada persona que utiliza Mokey, de forma que se pueda hacer un seguimiento. Los datos se almacenan en el directorio raíz de la sesión en un archivo con extensión *txt* renombrado con el mismo nombre de la sesión con el objetivo de poder realizar su carga cuando se inicie de nuevo la misma sesión.

Esta información es utilizada posteriormente en la exportación de los registros de uso. Es posible modificar estos campos siempre que se inicie en la sesión del usuario requerido pulsando con el ratón en el botón “Save”. En caso de no querer rellenar nada se guardan datos por defecto como los que se muestran en la figura 3.11.

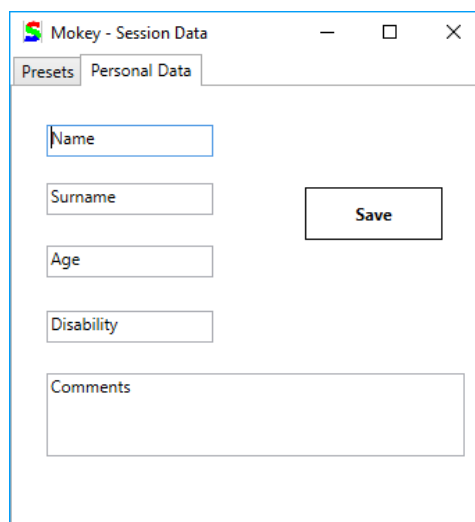


Figura 3.11: Pantalla de edición de datos personales de la sesión personal de Mokey.

Una vez guardados los datos en la pestaña de datos personales, para continuar a la siguiente ventana es necesario volver a la pestaña de *presets* y pulsar en el botón “Start”. Hecho esto, Mokey carga la interfaz de configuración principal con los datos almacenados en el *preset* seleccionado o con los valores por defecto si se trata de una sesión completamente nueva. Esta ventana principal es la que se muestra en la figura 3.12.

Configuración de movimientos

Esta ventana principal se compone de dos pestañas, una primera mostrada en la figura 3.12, relativa a la configuración de los doce movimientos predefinidos y otra pestaña adicional dedicada a la configuración de los cuatro movimientos grabados, mostrada en la figura 3.13. En la ventana principal de Mokey, en la pestaña “Basic gestures”, se muestran todos los controles y objetos que permiten la configuración de cada uno de los doce movimientos predefinidos, así como sus parámetros asociados: *displacement*, *delay* y tecla definida en el movimiento. Adicionalmente, en esta misma pestaña se configura el modo “Wheelchair” que se describe en el apartado 3.4.1, y se configura la ventana o aplicación sobre la que se quiere que actúe Mokey. Además, es posible cargar configuraciones previamente guardadas pertenecientes a otras sesiones con el botón “Load configuration”. En la barra de título se muestra el logo junto con el nombre del programa y la versión de compilación actual además de las acciones de minimizar y cerrar programa.

El diseño de esta pestaña está dividido en cuatro bloques claramente diferenciados con recuadros negros acompañados con varios elementos fuera de dichos recuadros. En la parte superior de la pestaña, se encuentra un texto descriptivo con el nombre del usuario de la sesión que ha sido establecida al inicio de Mokey y posteriormente se da lugar a los dos bloques principales del programa. El primer bloque, situado en la parte superior izquierda, es el relativo a la selección de movimientos asociados a partes del esqueleto de Kinect que es posible configurar junto con el valor del parámetro rango o *displacement* asociado a cada uno de ellos. El segundo bloque situado en la parte derecha superior es el relativo a la configuración de la tecla que se quiere asociar a cada movimiento configurado, así como el ajuste del retardo o *delay* relativo a la separación entre eventos que simula Mokey con cada movimiento. Los bloques inferiores hacen referencia, a la activación del modo “Wheelchair” en la parte izquierda, y a la selección de la aplicación o programa que va a ser usado a través de movimientos gracias a Mokey. Por debajo de estos cuatro bloques se muestran el botón “Load configuration” para cargar configuraciones previamente guardadas, así como el nombre de los autores en la parte central y el logo del centro CITSEM en la parte derecha.

El diseño de esta ventana ha sido pensado para facilitar visualmente la configuración de Mokey. Se ha establecido el orden de los movimientos predefinidos de forma que se alternan los movimientos de la zona derecha del cuerpo con los de la zona izquierda debido a que, a la hora de utilizar dos movimientos, es inmediato usar el mismo movimiento asociado a los dos lados, derecha e izquierda. Por ejemplo, para configurar Mokey para el uso del Tetris, es inmediato que para mover la pieza a la derecha se emplee el brazo derecho y para mover la pieza a la izquierda se emplee el brazo izquierdo. Por el contrario, no es inmediato configurar la pierna derecha para mover la pieza a un lado y el brazo derecho para mover la pieza al otro. La separación en módulos está pensada para separar la configuración referente a los gestos en sí, con la configuración de la tecla asociada a cada gesto. Todos los objetos relativos al mismo movimiento están alineados horizontalmente para facilitar su configuración. Si tan solo se selecciona un movimiento en los cuadros de selección de la izquierda, no es necesario

configurar otros parámetros que no sean los relativos a ese movimiento activado dispuestos en línea horizontal con el cuadro de selección.

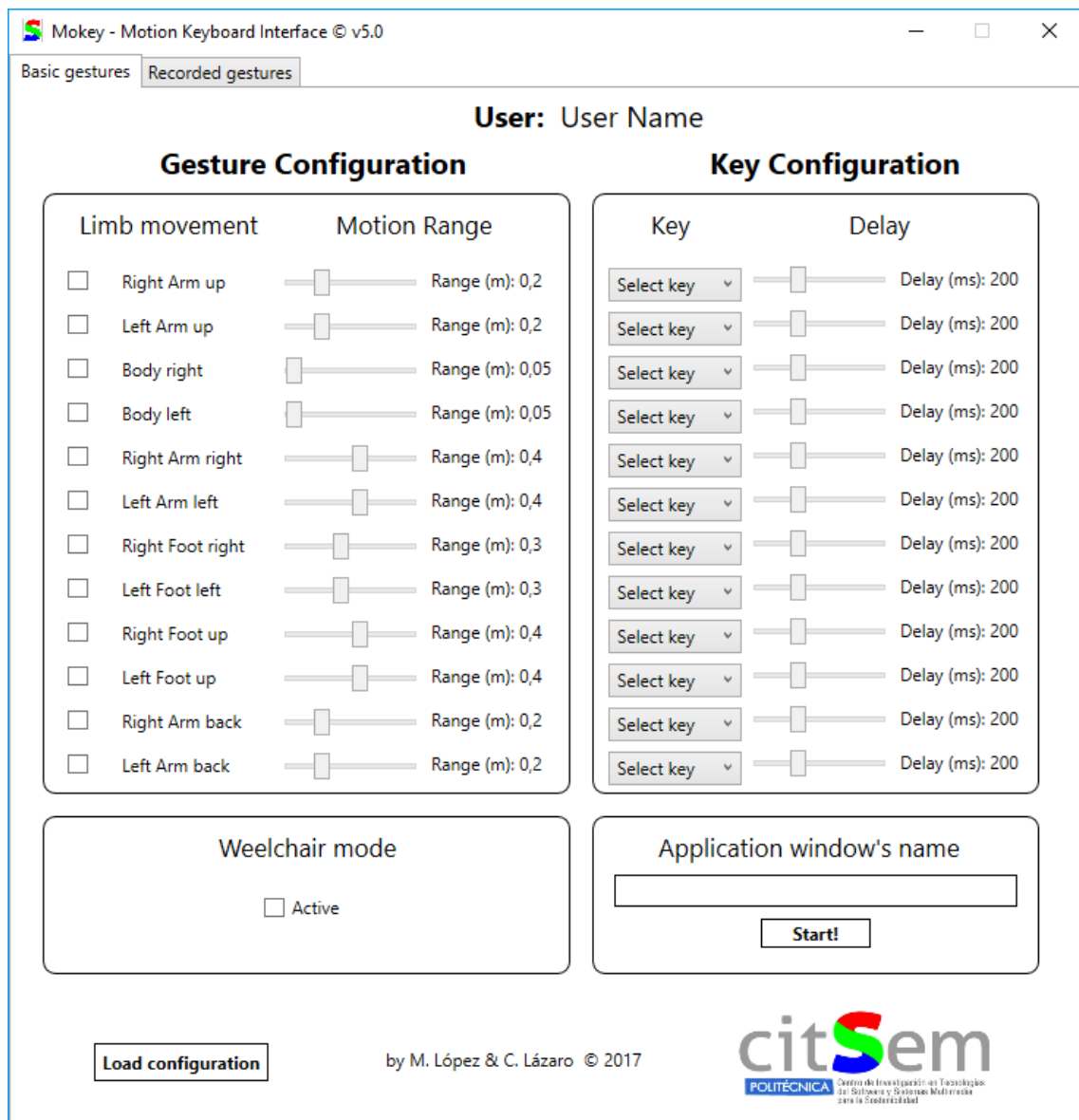


Figura 3.12: Pantalla principal de configuración de Mokey.

El primer módulo relativo a la configuración de gestos está compuesto por dos textos descriptivos de los botones y objetos que se sitúan debajo de ellos y que se refieren a los doce movimientos predefinidos configurables. Cada uno de estos doce movimientos se compone de un cuadro de selección para establecer la activación de cada uno de ellos, un texto descriptivo de cada movimiento y el ajuste dedicado al parámetro de *displacement*, mostrado en la interfaz en la sección “Motion Range”, compuesto de una barra deslizable y un texto informativo de la posición de la barra, entre 0 y 0.7 metros.

El segundo módulo dedicado a la configuración de las teclas está formado por dos textos descriptivos de los objetos de configuración. Debajo del texto “Key” se encuentran los cuadros de selección de evento de teclado de cada movimiento, pudiendo ser cualquier tecla del teclado

español¹ de entre las siguientes: cualquier tecla de letra del alfabeto, teclas de función más comunes (F1, F2 o F5), tecla de *enter*, barra espaciadora, o cualquiera de los cuatro cursores. También es posible configurar en este desplegable las acciones de “Llamar Skype” y “Colgar Skype” relativas a acciones del programa Microsoft Skype para hacer videoconferencias y llamadas telefónicas por internet. En la parte derecha se sitúa el control deslizante del ajuste de retardo o *delay* de cada movimiento acompañado del texto informativo de la posición del deslizante entre 0 y 600 milisegundos.

El tercer módulo relativo a la configuración del “Modo Wheelchair” tan solo tiene el texto del título y un cuadro de selección para determinar su activación o no. El cuarto módulo está compuesto del texto del título, un cuadro de introducción de texto destinado al nombre de la ventana de la aplicación que se utiliza con Mokey y por último el botón de “Start” que da lugar al inicio de Mokey con la configuración establecida. Una vez pulsado este botón, Mokey crea los hilos necesarios en relación a los movimientos activos, configura los parámetros acordes a la información recogida en la interfaz y envía las instrucciones para que la aplicación cuyo título de ventana está definido en el módulo correspondiente pase a primer plano. Esto último ocurre una vez el usuario se sitúe frente a la cámara Kinect con el objetivo de mejorar la comodidad y experiencia de usuario. La implementación de todos los algoritmos se puede ver detalladamente en [1].

La segunda pestaña, que se puede observar en la figura 3.13, es relativa a la configuración de los cuatro movimientos grabados que permite Mokey. Cada uno de los cuatro movimientos forma parte de un módulo diferenciado, pero similar a los otros tres, acotado por un recuadro negro. Cada uno de los módulos está formado por un texto con el número de movimiento que hace referencia, un cuadro de selección para su activación, un cuadro con la selección referente a la parte del cuerpo que se pretende grabar y, al igual que en los movimientos de la primera pestaña, una lista desplegable con la tecla que se quiere asociar al movimiento y el parámetro *delay* asociado al mismo. Por último, cada uno de los cuatro movimientos tiene asociado un botón con el texto “Record” destinado a grabar el movimiento. En el apartado 3.5 se describe con detalle este aspecto de la grabación de movimientos. Esta pestaña tan solo es de configuración, para ejecutar Mokey es necesario volver a la primera pestaña “Basic gestures” y pulsar en el botón “Start!”.

Una vez ejecutado Mokey tras rellenar el cuadro de texto con el nombre de la ventana de la aplicación que se quiere controlar y hacer clic en el botón descrito, la Interfaz Natural Mokey empieza a detectar los movimientos acordes a la configuración previa en la interfaz. Una vez el usuario se coloca en frente de la cámara Kinect, Mokey trae a primer plano la aplicación controlada y comienza la simulación de eventos de teclado según lo configurado. Mokey, además, permite la configuración en tiempo real de los parámetros de *delay* o *displacement* así como la tecla simulada por cada movimiento sin la necesidad de parar el programa o volver a ejecutar ninguna instrucción adicional. Esta característica es muy útil a la hora de configurar con una mayor precisión cada uno de los ajustes y parámetros de los movimientos establecidos por el usuario.

¹ Definido como teclado QWERTY con alfabetización nacional de España.

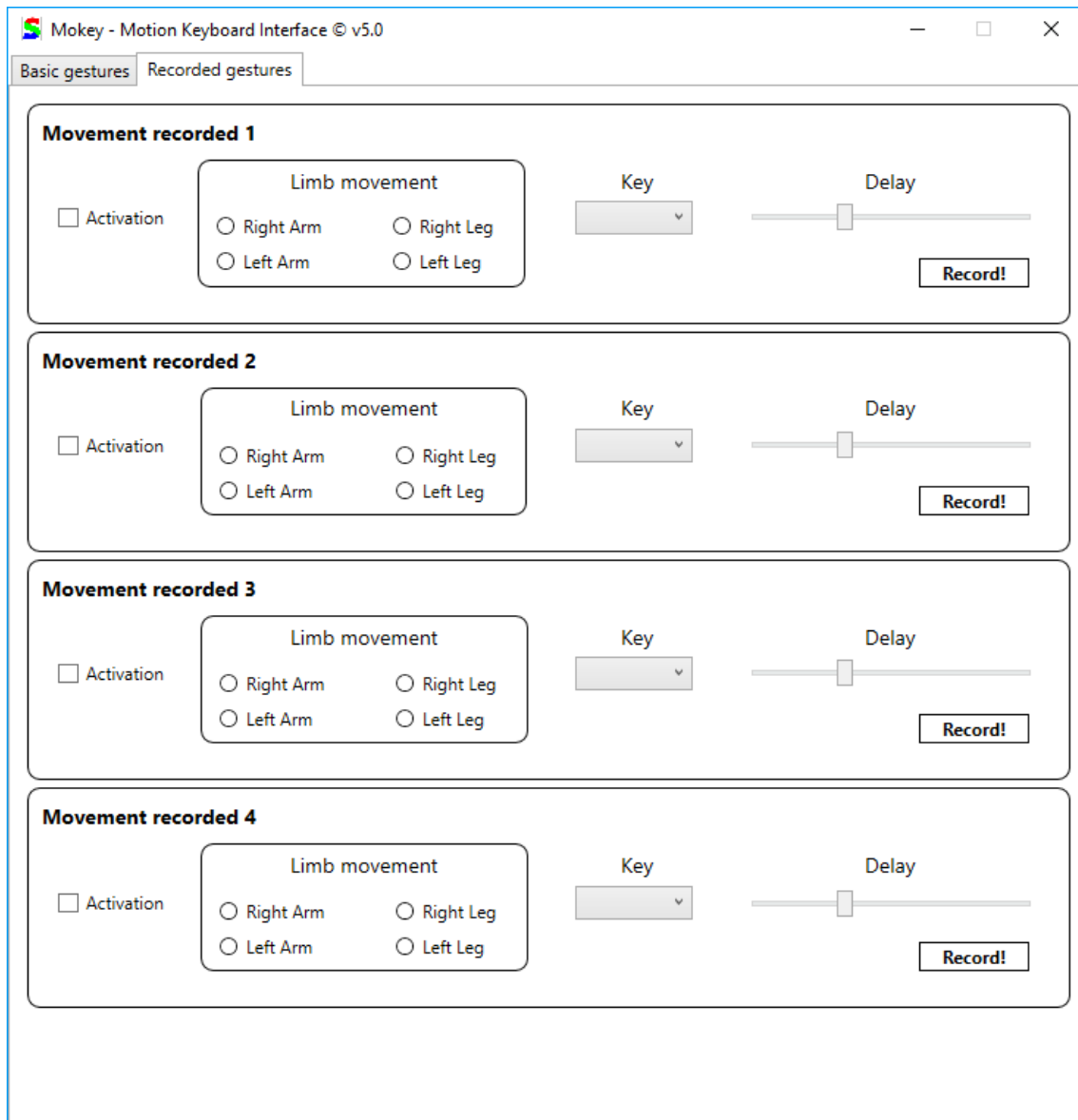


Figura 3.13: Pantalla principal de configuración de movimientos de grabación de Mokey.

Guardado de valores por defecto (*preset*)

Una vez el usuario finaliza todas las acciones con la aplicación o programa en cuestión y quiere cerrar la sesión de Mokey, al cerrar pulsando sobre el botón de cerrar de Windows en la ventana principal mostrada en la figura 3.12, Mokey hace un salto a una nueva ventana de guardado de *presets*. Esta ventana que se muestra en la figura 3.14, está destinada al guardado de la configuración utilizada por última vez en Mokey. Está formada por un texto en la parte superior que pregunta si se quiere guardar o no la configuración y a continuación dos cuadros de textos para escribir sobre ellos ya rellenos con valores por defecto. El primer cuadro de texto hace referencia al nombre con el que se guarda el archivo de configuración y el segundo cuadro está destinado a guardar comentarios que el usuario considere importantes o de relevancia. Por último, se muestran dos botones, el primero de ellos, el botón “Yes”, guarda el archivo de configuración en un fichero de texto con extensión *txt* con la información recogida en la interfaz y en los dos campos de la misma ventana. Por otro lado, el botón “No” descarta toda la

configuración, cierra el programa y finaliza todos los procesos de Mokey. De esta forma, no es necesario volver a establecer desde cero aquella configuración de los movimientos seleccionados para un uso concreto

Una vez establecido los campos de nombre de fichero y de los comentarios opcionales del usuario, Mokey almacena la última configuración de la interfaz dentro del directorio “configs” en la ruta de la sesión iniciada por última vez. Para volver a cargar en la interfaz la configuración guardada, es necesario iniciar el programa con la misma sesión en la cual fue guardado el archivo y seleccionar el archivo en la ventana de carga de archivos de configuración mostrada en la figura 3.10.

La información recogida en los dos cuadros de texto mostrados en la figura 3.14, es la misma que, en posteriores inicios de sesión bajo el mismo usuario, se muestran en la ventana de carga de configuraciones almacenadas. Un ejemplo de ello se observa en la figura 3.10, donde se listan las configuraciones almacenadas previamente con el nombre de archivo y comentarios guardados al finalizar Mokey.

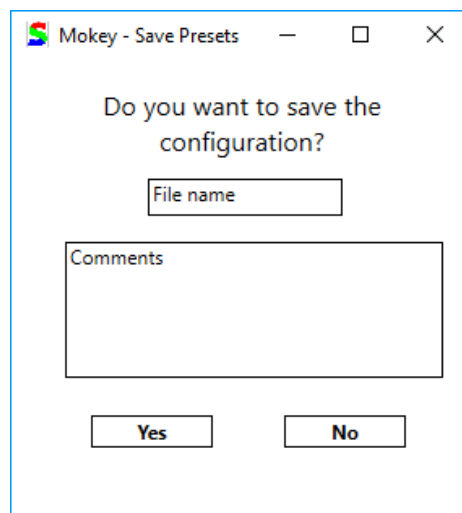


Figura 3.14: Pantalla de fin de sesión y guardado de Mokey.

Los archivos de configuración guardados no son exclusivos. Es posible extraerlos con el navegador de archivos de Windows y almacenarlos en cualquier directorio personal para importarlo desde el botón “Load configuration” de la ventana principal en otras sesiones diferentes a la sesión en la que fue guardado.

3.3.1. Elementos

Para el desarrollo de la interfaz gráfica, Visual Studio confecciona un archivo por cada ventana con extensión *.xaml* con todos los objetos de la interfaz, sus parámetros, ajustes y eventos asociados. Por ello es posible editar dichos objetos desde el código, como la posición exacta en el lienzo de la interfaz gráfica, el tamaño, nombre, etc.

Cada una de las ventanas está programada en un archivo *xaml* diferente bajo la clase *Window* de la librería de *System.Windows* [30] de Visual Studio. Cada uno de los elementos de la interfaz pertenecen a esta misma librería, por lo que es necesario referenciarla en el proyecto de Visual Studio para que sean reconocidos y compilados todos los objetos. Todos los elementos de la interfaz pertenecen a esta misma librería.

Los objetos de la interfaz gráfica, pertenecientes a la librería *System.Windows*, que se han empleado en la programación de todas las ventanas de Mokey, con su forma de declararlos en el proyecto, son los siguientes:

- **TabControl y TabItem**

Hacen referencia a las pestañas que se configuran en cada ventana, pudiendo a su vez introducir multitud de objetos y elementos en cada una de ellas. El cambio entre pestañas se realiza en la parte superior de la ventana, debajo de la barra de título de la misma. Para hacer uso de este tipo de objetos, es necesario declarar inicialmente un objeto “TabControl” y, a continuación, tantos objetos “TabItem” como pestañas tenga la ventana. Es posible asignar multitud de ajustes como color, título y tamaño a cada una de las pestañas.

```
<TabControl /Atributos de pestañas/
  <TabItem Header="Nombre pestaña 1">
    /Contenido pestaña 1/
  </TabItem>
  <TabItem Header="Nombre pestaña 2">
    /Contenido pestaña 2/
  </TabItem>
</TabControl>
```

- **Border**

Este objeto está formado por un cuadro libre de texto de tamaño establecido por el usuario que es posible ubicar en cualquier parte del lienzo de la ventana o pestaña correspondiente. Permite personalización de color, línea o forma. Este elemento es empleado para delimitar las zonas o módulos de la interfaz gráfica.

```
<Border /Atributos del objeto/ />
```

- **TextBlock**

Se trata de un cuadro de texto no editable en el cual es personalizable la fuente, color, formato, tamaño o ubicación del mismo. En Mokey es empleado en cada uno de los carteles, rótulos y textos informativos de la interfaz, por ejemplo, en los carteles descriptivos de los movimientos en la interfaz gráfica. Es posible determinar su contenido a través del código fuente del programa.

```
<TextBlock x:Name="Nombre objeto" Text="Texto a mostrar" /Atributos objeto/ />
```

- **TextBox**

Este objeto es similar al anterior, al objeto “TextBlock”, pero en este caso se trata de un cuadro de texto que sí permite su edición y la escritura en el mismo. Es posible redimensionarlo y establecer el formato del texto mostrado. En Mokey es el objeto empleado para añadir desde la interfaz gráfica información de entrada como los datos personales, el nombre de la sesión o los comentarios al guardar un archivo de configuración. La información recogida en estos elementos se almacena en cadenas de caracteres en variables del código de programa.

```
<TextBox Name="Nombre objeto" Text="Texto del cuadro" /Atributos objeto/ />
```

- **ComboBox y ListBoxItem**

Al igual que con los objetos “TabControl” y “TabItem”, este elemento se programa bajo una estructura similar. En este caso se trata de un cuadro de texto editable similar a “TextBox” pero en este caso con un desplegable de valores predefinidos. Estos valores predefinidos son los elementos definidos como “ListBoxItem”. En Mokey son los objetos empleados para los controles de elección de evento de tecla para cada uno de los movimientos. El valor seleccionado se almacena en una variable del programa principal.

```
<ComboBox x:Name="Nombre elemento" /Atributos/ >
  <ListBoxItem Content="Contenido elemento 1"/>
  <ListBoxItem Content="Contenido elemento 2"/>
  <ListBoxItem Content="Contenido elemento n"/>
</ComboBox>
```

- **List Box**

Este elemento hace referencia a listas de elementos listados en un cuadro de texto y seleccionables. Es similar a un objeto del tipo “ComboBox” pero sin desplegable, mostrado todo en forma de lista dentro de un cuadro establecido. En Mokey se emplea para mostrar la lista de archivos de configuración guardados en una misma sesión en la ventana de la figura 3.10. En este caso, el resultado de la selección también se almacena en una variable del código de programa, donde en el caso de Mokey, también se realiza la búsqueda de archivos de configuración en un método referenciado en el código, es decir, en un evento de Visual Studio.

```
<ListBox x:Name="Nombre Objeto" SelectionChanged="NombreMétodoEvento" /Atributos/ />
```

- **CheckBox**

Un elemento del tipo “CheckBox” es un objeto muy recurrido a la hora de precisar de cuadros de selección que establezca el valor de una variable de tipo *boolean* en el código de programa. En el caso de Mokey, estos elementos son utilizados para que el usuario determine qué movimientos de todos los disponibles se activan, así como un modo o característica. Se trata de un cuadro con dos posibles estados controlados mediante un evento de Visual Studio: activado o desactivado.

```
<CheckBox x:Name="NombreElemento" Checked="NombreMétodoEvento" /Atributos/ />
```

- **Slider**

Este objeto hace referencia a una barra horizontal con un controlador deslizante para establecer un valor numérico comprendido entre dos magnitudes. Este intervalo, el número de decimales y el salto permitido, color, tamaño y ubicación son algunos de sus atributos principales. Es un objeto especialmente útil para personalizar el valor de variables numéricas con valores comprendidos en un rango mediante eventos. En Mokey, se emplean para establecer los valores de los parámetros de *delay* y *displacement* para cada movimiento.

```
<Slider x:Name="NombreElemento" ValueChanged="NombreMétodoEvento" /Atributos/ />
```

- **Button**

Un elemento de tipo “Button” hace referencia a una superficie delimitada por un recuadro, donde al pulsar con el ratón del ordenador, el evento de Visual Studio asociado ejecuta una rutina predefinida en el código de programa. En Mokey se emplean objetos de este tipo para ejecutar acciones, guardar datos, avanzar entre ventanas de configuración, etc. Es un recurso muy utilizado en todas las interfaces gráficas.

```
<Button x:Name="NombreElemento" Click="NombreMétodoEvento" /Atributos/ />
```

- **Image**

Este tipo de objeto es insertado para representar imágenes previamente referenciadas en el proyecto de Visual Studio. Cuenta con personalización de tamaño, ubicación o tipo de marco como atributos principales. En Mokey, se ha empleado para las imágenes insertadas, como el logo del centro CITSEM.

```
<Image x:Name="NombreElemento" Source="NombreImagenReferenciada" /Atributos/ />
```

- **RadioButton**

De forma similar a un objeto de tipo “CheckBox”, este objeto hace referencia a un elemento circular que establece la activación o no de acciones o funciones de programa. Es empleado para disponer de diferentes opciones, asociadas al mismo grupo, con una sola selección posible de forma simultánea y gobernado mediante eventos en forma de métodos en el código de programa. Es usado en Mokey al seleccionar el miembro a grabar en la pestaña de grabación de movimientos de la figura 3.13, en el cuadro “Limb Movement”.

```
<RadioButton x:Name="NombreElemento" Checked="Evento Activado" Unchecked="Evento Desactivado" GroupName="NombreGrupo" /Atributos/ />
```

3.4. Movimientos configurables

En este apartado se describe cada uno de los doce movimientos predefinidos que es posible configurar en Mokey. Durante la realización del proyecto, este abanico de movimientos ha ido evolucionando y cambiando para adaptarse lo máximo posible a las necesidades y uso del usuario medio teniendo en cuenta un gran número de tipos de discapacidad. Por ello, Mokey ha sido diseñado buscando la máxima versatilidad y adaptabilidad a cualquier tipo de situación. Los doce movimientos predefinidos, según están ordenados en la interfaz de configuración de Mokey, se puede observar en la tabla 3.1.

Cada uno de los movimientos está diseñado partiendo de una situación de reposo como se muestra en la figura 3.15, tanto en posición erguida como en posición sentada. A partir de este estado de reposo, Mokey reconoce, mediante los algoritmos descritos en [1], qué movimientos se realizan en cada instante por el usuario para seguidamente ejecutar el evento de teclado configurado en el mismo. Mokey trabaja en tiempo real leyendo las coordenadas relativas (X, Y y Z) de los puntos del esqueleto de Kinect del usuario que se encuentre frente a la cámara. Al almacenar en tiempo real dichas coordenadas en variables del sistema, Mokey compara diferencias relativas entre las coordenadas de ciertas articulaciones en un eje de coordenadas concreto, dependiendo del movimiento en cuestión, para evaluar si se superan los umbrales previamente configurados mediante el parámetro *displacement*. Si se cumple la condición del

algoritmo de cada movimiento, Mokey ejecuta el evento de teclado o acción configurada. A lo largo de este apartado se detallan cada uno de estos movimientos configurables en Mokey y se explica cómo funcionan.

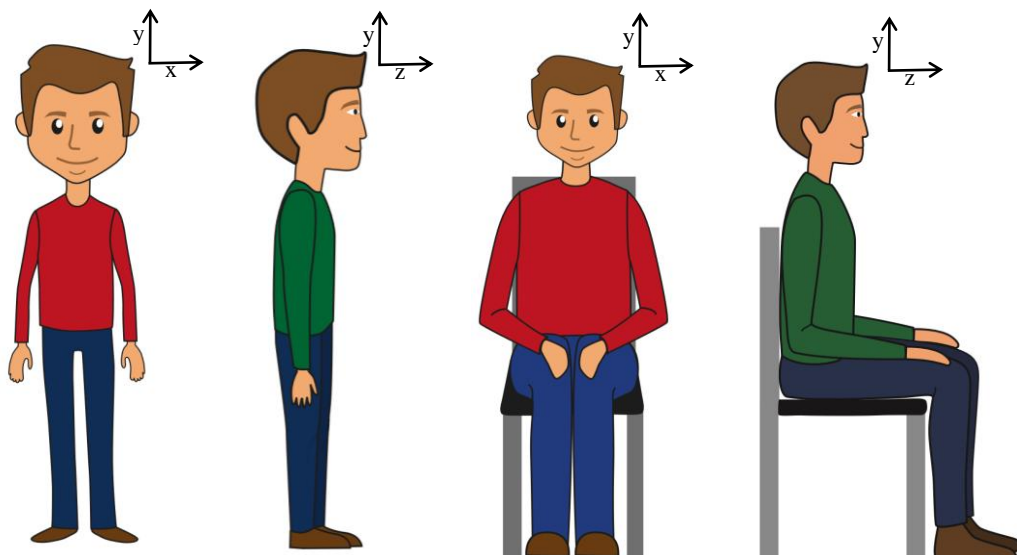


Figura 3.15: Estado de reposo del usuario del cual parten todos los movimientos configurables, tanto de pie como sentado.

Cada uno de los doce movimientos configurables de forma predefinida en Mokey, tiene su propio algoritmo de comparación para determinar si un movimiento determinado se realiza o no. Cada algoritmo compara dos huesos o puntos del esqueleto proporcionado por las funciones de la SDK, como se muestra en la figura 2.4. Como se puede observar en la tabla 3.1, para cada movimiento, uno de estos huesos se toma como referencia y el otro como el hueso indicador del movimiento. Según las coordenadas espaciales de cada hueso, cada algoritmo evalúa, teniendo en cuenta el parámetro *displacement*, si el movimiento se cumple o no. Estos algoritmos y la forma de obtener las coordenadas de cada hueso están explicados con detalle en [1].

Tabla 3.1: Relación de movimientos con los huesos implicados y eje de interés

Movimientos	Hueso principal	Hueso referencia	Eje coordenadas
1. Brazo derecho hacia arriba	<i>Hand_Right</i>	<i>Spine</i>	Y
2. Brazo izquierdo hacia arriba	<i>Hand_Left</i>	<i>Spine</i>	Y
3. Tronco hacia la derecha	<i>Head</i>	<i>Spine</i>	X
4. Tronco hacia la izquierda	<i>Head</i>	<i>Spine</i>	X
5. Brazo derecho hacia la derecha	<i>Hand_Right</i>	<i>Spine</i>	X
6. Brazo izquierdo hacia la izquierda	<i>Hand_Left</i>	<i>Spine</i>	X
7. Pie derecho hacia la derecha	<i>Foot_Right</i>	<i>Spine</i>	X
8. Pie izquierdo hacia la izquierda	<i>Foot_Left</i>	<i>Spine</i>	X
9. Pie derecho hacia delante	<i>Foot_Right</i>	<i>Spine</i>	Z
10. Pie izquierdo hacia delante	<i>Foot_Left</i>	<i>Spine</i>	Z
11. Brazo derecho hacia atrás	<i>Hand_Right</i>	<i>Spine</i>	Z
12. Brazo izquierdo hacia atrás	<i>Hand_Left</i>	<i>Spine</i>	Z

Uno de los objetivos de Mokey es el de conseguir la máxima adaptabilidad al mayor número de usuarios posible. Para ello se han diseñado los doce movimientos de forma que abarquen los gestos más comunes y sencillos con el objetivo de poder manejar el mayor número de aplicaciones posible. A continuación, se detalla cada uno de ellos.

- **Movimiento 1: Brazo derecho hacia arriba**

Este primer movimiento se representa en la parte izquierda de la figura 3.16 y hace referencia al movimiento de levantar verticalmente el brazo derecho, hueso *Hand_Right* hacia arriba. Es quizá el gesto más cómodo y el que puede utilizar mayor número de personas. En el momento en el que la mano es levantada, la coordenada del punto *Hand_Right* relativa al eje Y aumenta, haciendo más grande a su vez la diferencia respecto a *Spine*, punto situado en la columna vertebral a la altura del esternón. En el estado de reposo esta diferencia es negativa al estar la mano por debajo. Si esta diferencia es mayor que el parámetro *displacement* configurado, el movimiento se toma como realizado. Para la realización de este movimiento no es necesario mantener el brazo rígido.

- **Movimiento 2: Brazo izquierdo hacia arriba**

Este movimiento predefinido es similar al movimiento 1 descrito anteriormente, pero, en este caso, realizado con la mano izquierda, el hueso *Hand_Left*. Se puede observar este movimiento en la figura 3.16 derecha.

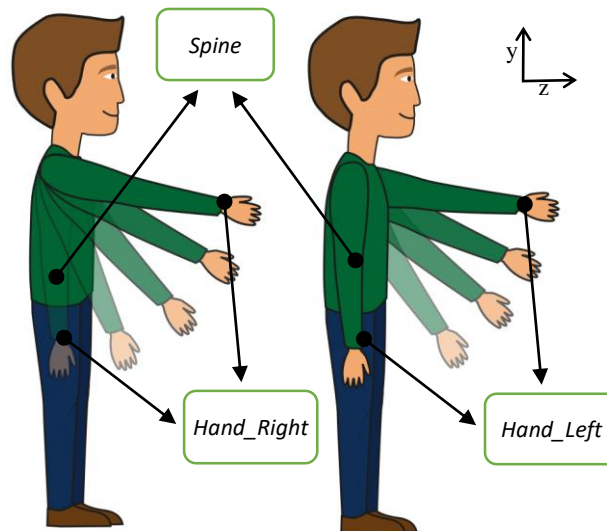


Figura 3.16: Representación gráfica del movimiento 1: Brazo derecho hacia arriba (izquierda) y movimiento 2: Brazo izquierdo hacia arriba (derecha).

- **Movimiento 3: Tronco hacia la derecha**

El tercer movimiento hace referencia a la inclinación del tronco del cuerpo hacia el lado derecho del mismo. En este caso la comparación entre coordenadas se realiza con el punto *Head* relativo a la cabeza frente al punto *Spine* del cuerpo detectado en el eje X de coordenadas. Este movimiento está representado en la figura 3.17 derecha y está

especialmente dirigido a aquellas personas con movilidad reducida en brazos y para usuarios en silla de ruedas debido a la importancia de ejercitar la musculación del tronco.

Es importante señalar que este movimiento funciona incluso en silla de ruedas, por lo que es un recurso de vital importancia para multitud de personas con problemas de movilidad en brazos y piernas.

Este movimiento sustituye al movimiento de “bajar cabeza/agacharse” expuesto en [1] por M. López. La evolución del programa en el periodo entre una memoria y otra ha llevado a la supresión de este movimiento debido a su escaso uso y la alta capacidad de procesamiento que requería resultado de la calibración previa que necesitaba. Por ello, se presenta un movimiento más útil, en especial para usuarios en silla de ruedas, según se recoge en el capítulo de resultados, y una mejor experiencia de usuario debido a la supresión del sistema de calibración necesario hasta ese momento en el movimiento anterior tal y como se detalla en [1].

- **Movimiento 4: Tronco hacia la izquierda**

Al igual que en el movimiento 3, este cuarto movimiento hace referencia a la inclinación del tronco del cuerpo, en este caso, hacia la izquierda del mismo. Este movimiento se puede observar en la figura 3.17 izquierda. De igual forma, se compara la cabeza con la posición de la parte central de la columna vertebral.

Al igual que ocurre con el movimiento 5, este movimiento 4 sustituye al movimiento de “levantar cabeza/ponerse de puntillas” explicado en la memoria [1] de M. López suprimiendo también el proceso de calibración previa expuesto en el mismo. Este proceso de calibración era exclusivo para estos dos movimientos sustituidos por lo que ya no es necesario en el programa. Los dos nuevos movimientos introducidos realizan comparaciones de las coordenadas relativas en cada instante de los dos puntos afectados -*Head* y *Spine*- de la misma forma que ocurre con los restantes 10 movimientos predefinidos en el programa.

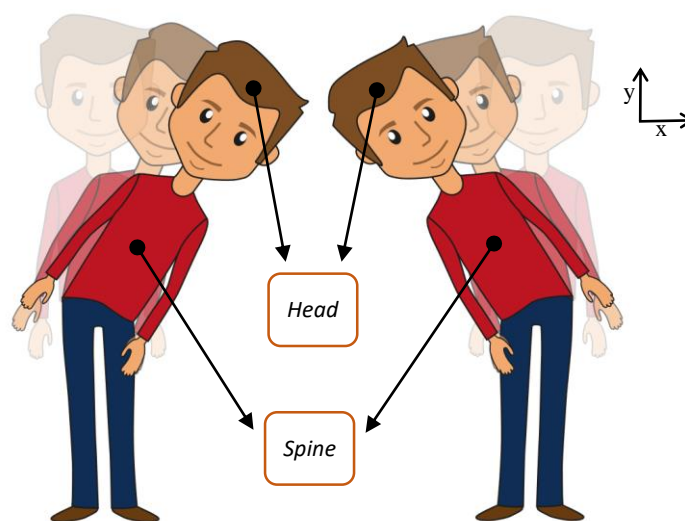


Figura 3.17: Representación gráfica del movimiento 3: Tronco hacia la derecha (derecha) y movimiento 4: Tronco hacia la izquierda (izquierda).

- **Movimiento 5: Brazo derecho hacia la derecha**

En esta ocasión, este movimiento es mostrado en la figura 3.18 derecha y hace referencia al movimiento de forma lateral del brazo derecho a ese mismo lado, en el eje de coordenadas X. Los huesos que cobran principal relevancia son, al igual que en el movimiento 1, los de la mano derecha, tomado como principal, y el punto detectado como *Spine*.

Es importante señalar que este quinto movimiento de mover el brazo derecho a la derecha puede entrar en conflicto con el movimiento 1 de levantar el brazo derecho hacia arriba. Esto es debido a que, moviendo el brazo al lateral, se suele levantar respecto al punto de referencia *Spine*. Para que Mokey no entre en conflicto si se seleccionan los dos movimientos de forma simultánea, se establece este movimiento 5 prioritario frente al movimiento 1. De esta forma se detecta con seguridad si el gesto de levantar el brazo es hacia un lateral o si por el contrario es hacia delante. La forma de establecer la prioridad es evaluando dos condiciones en el algoritmo relativo al movimiento:

- ✓ Si se cumple *MOV1*, pero no *MOV5*, entonces ejecutar *EVENTO MOV1*
- ✓ Si se cumple *MOV5* -y por defecto *MOV1*-, entonces ejecutar *EVENTO MOV5*

- **Movimiento 6: Brazo izquierdo hacia la izquierda**

El sexto movimiento, mostrado en la figura 3.18 izquierda, hace referencia al movimiento del brazo izquierdo a la izquierda. Es similar al movimiento 5 pero en este caso entra como hueso principal el de la mano izquierda para comparar sus coordenadas respecto a *Spine*. Del mismo modo que ocurre con los movimientos 1 y 5, para evitar un conflicto de este movimiento 6 con el movimiento 2 de levantar el brazo izquierdo hacia delante, este movimiento 6 se establece como prioritario frente al número 2.

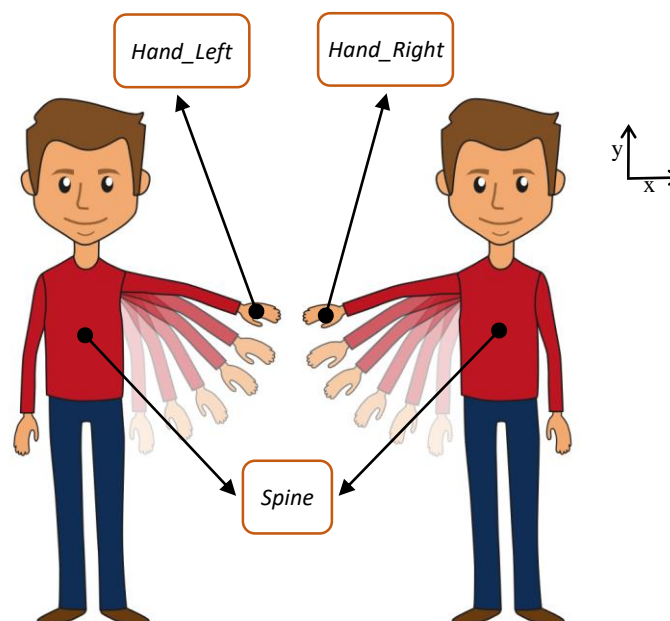


Figura 3.18: Representación gráfica del movimiento 5: Brazo derecho hacia la derecha (derecha) y movimiento 6: Brazo izquierdo hacia la izquierda (izquierda).

- **Movimiento 7: Pie derecho hacia la derecha**

Este séptimo movimiento hace referencia al movimiento del pie derecho a la derecha de forma similar al movimiento 5 de mover el brazo a la derecha. En este caso el hueso principal a comprar es el pie derecho, denominado por Kinect como *Foot_Right*, con el punto *Spine* del esqueleto según el eje de coordenadas X. La representación de este movimiento se puede ver en la figura 3.19 derecha.

- **Movimiento 8: Pie izquierdo hacia la izquierda**

Al igual que con el movimiento 7, este octavo movimiento hace referencia al desplazamiento lateral del pie, en este caso del pie izquierdo, hueso *Foot_Left*, a la izquierda en el eje X de coordenadas. Este movimiento se puede observar en la parte derecha de la figura 3.19.

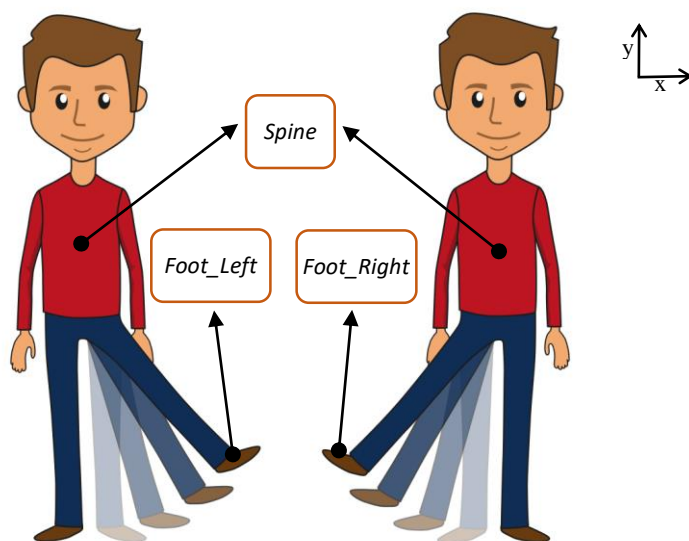


Figura 3.19: Representación gráfica del movimiento 7: Pie derecho hacia la derecha (derecha) y movimiento 8: Pie izquierdo hacia la izquierda (izquierda).

- **Movimiento 9: Pie derecho hacia delante**

Como se puede observar en parte izquierda de la figura 3.20, el movimiento número 9 hace referencia al desplazamiento del pie derecho, hueso *Foot_Right*, hacia delante del cuerpo. De esta forma, si el algoritmo relativo al movimiento detecta un desplazamiento mayor que el umbral, o valor de *displacement*, en el eje Z respecto a *Spine*, Mokey ejecuta el evento correspondiente de teclado siempre que este desplazamiento sea de acercamiento a la cámara Kinect.

- **Movimiento 10: Pie izquierdo hacia delante**

El décimo movimiento hace referencia, de forma similar al movimiento 9, al desplazamiento del pie izquierdo, punto del esqueleto *Foot_Left*, hacia delante. Está mostrado en la parte derecha de la figura 3.20.

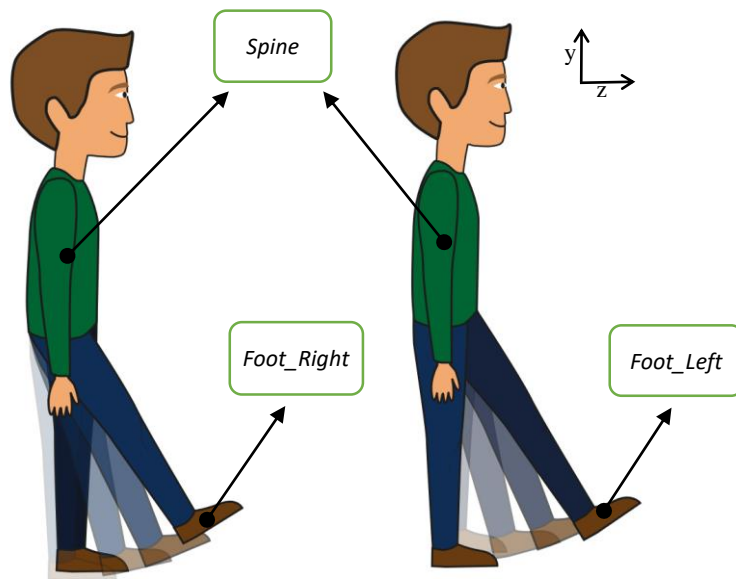


Figura 3.20: Representación gráfica del movimiento 9: Pie derecho hacia delante (izquierda) y movimiento 10: Pie izquierdo hacia delante (derecha).

- **Movimiento 11: Brazo derecho hacia atrás**

El movimiento 11, mostrado en la figura 3.21 izquierda, ofrece grandes posibilidades para aplicaciones que requiera numerosos movimientos debido a que no es un gesto muy natural en una persona y puede ser destinado a una tecla utilizada con poca frecuencia o para casos puntuales.

Se trata del movimiento del brazo derecho, punto *Hand_Right* de Kinect, hacia detrás del cuerpo. Para este movimiento, el hueso principal de referencia es comparado con las coordenadas del punto *Spine* de Kinect evaluando un aumento de la diferencia entre ambos puntos en el eje Z, el que establece la profundidad en la detección de Kinect. La representación se puede observar en la figura 3.21 izquierda.

- **Movimiento 12: Brazo izquierdo hacia atrás**

El último movimiento dentro de los predefinidos es el número 12. Este movimiento está representado en la figura 3.21 derecha y es igual que el número 11 pero, en esta ocasión, comparando como hueso principal el de la mano izquierda, *Hand_Left*.

Al no precisar ninguno de estos 12 movimientos del proceso calibración detallado en [1], se decide suprimirlo del programa. Este resultado aporta una configuración más rápida y una ejecución de Mokey instantánea al pulsar el botón “Play!” de la interfaz principal de configuración del programa.

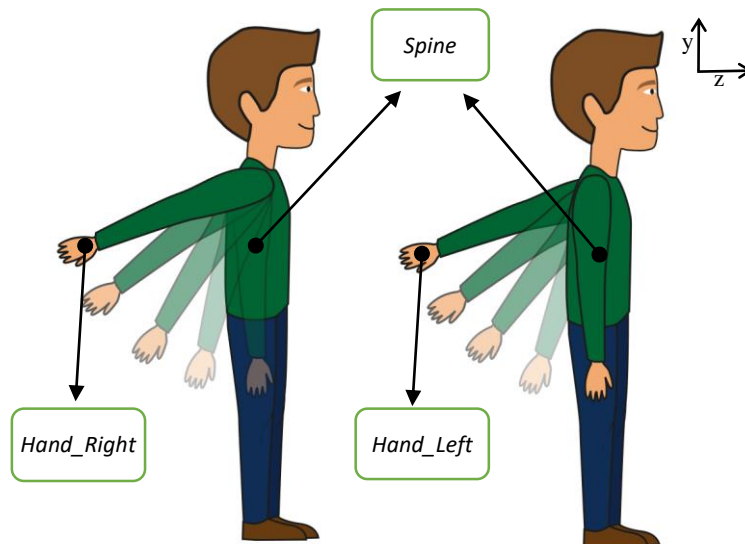


Figura 3.21: Representación gráfica del movimiento 11: Brazo derecho hacia atrás (izquierda) y movimiento 12: Brazo izquierdo hacia atrás (derecha).

3.4.1 Modo *Wheelchair*

Para hacer más accesible Mokey, y sus virtudes y funciones, a usuarios con movilidad reducida o a personas con discapacidad que están obligadas a permanecer en silla de ruedas se ha pensado en un modo de configuración durante el desarrollo de Mokey especialmente diseñado para hacer la experiencia de usuario lo más satisfactoria posible.

Este modo se activa en la pantalla principal de Mokey mostrada en la figura 3.12 y consta únicamente de un cuadro de selección de activación como se puede observar en la figura 3.22. Este modo no afecta, de cara al usuario, a la hora de la configuración de los movimientos y parámetros para hacer uso de Mokey, pero sí que afecta de forma interna en la forma en la que Mokey gestiona los algoritmos de cada uno de los movimientos.

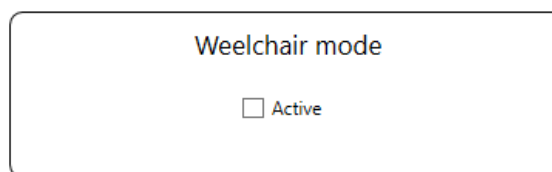


Figura 3.22: Control de activación del “Modo *Wheelchair*” en la pantalla principal de Mokey.

Como ya se ha detallado en el apartado 3.4, Mokey utiliza como referencia para su comparación en todos los movimientos predefinidos el punto del esqueleto denominado por la SDK de Mokey como *Spine*. Este hueso equivale a la parte central de la columna vertebral, tal y como se muestra en la figura 2.4; posición que, al estar sentado, Kinect detecta con dificultad y con una alta probabilidad de error y de falsas detecciones si se comparan coordenadas de huesos como los pies o las rodillas. Esto es debido a que, cuando el usuario permanece sentado, las rodillas se posicionan por delante del punto de detección *Spine* dificultando así su correcta detección. Para subsanar esta dificultad, Mokey cuenta con un modo denominado “*Wheelchair mode*”.

En las primeras versiones de Mokey, este modo operaba de diferente forma a la actual. Al activarlo, este modo tenía como función deshabilitar los movimientos relativos a los pies y piernas dejando solamente activos los referentes a los brazos. Durante el desarrollo de Mokey se ve que esto no soluciona el problema, ya que hay usuarios que pese a estar sentados en silla de ruedas pueden mover con suficiente soltura los pies y las piernas. Además, se pierden numerosas posibilidades de configuración al eliminar posibles movimientos configurables.

Como alternativa a esta primera solución, se desarrolla un nuevo modo para usuarios que interactúan con el programa sentado. Este modo permite seleccionar cualquiera de los doce movimientos predefinidos de forma que no se ven disminuidas las capacidades de Mokey si se activa dicho modo sentado. Para este nuevo modo, se cambia la referencia de aquellos movimientos en los que se ven afectados los huesos de la cadera y las rodillas para mantener las mismas posibilidades de configuración. Los movimientos en los que se ven afectados dichos cambios son los que se muestran a continuación.

- Movimiento 7: Pie derecho hacia la derecha
- Movimiento 8: Pie izquierdo hacia la izquierda
- Movimiento 9: Pie derecho hacia delante
- Movimiento 10: Pie izquierdo hacia delante

En lugar de comparar el pie correspondiente con el hueso *Spine*, con esta configuración activada, Mokey compara, en cada hilo correspondiente, el pie en cuestión con las coordenadas de la rodilla del mismo lado, derecha o izquierda. Para estos movimientos, se lanza un hilo diferente en función de si está el cuadro de activación del modo sentado seleccionado o no. De esta forma quedan diferenciados los dos funcionamientos para poder seleccionar en tiempo real de funcionamiento de Mokey qué movimiento hilo se asocia a cada movimiento, el que toma como referencia el hueso *Spine*, o el que toma como referencia los huesos de las rodillas. En la tabla 3.2 se pueden observar dichos cambios en el hueso referencia de cada movimiento.

Tabla 3.2: Relación de movimientos con los huesos implicados en modo "Wheelchair".

Movimientos <i>Wheelchair Mode</i>	Hueso principal	Hueso referencia	Eje coordenadas
7. Pie derecho hacia la derecha	<i>Foot_Right</i>	<i>Knee_Right</i>	X
8. Pie izquierdo hacia la izquierda	<i>Foot_Left</i>	<i>Knee_Left</i>	X
9. Pie derecho hacia delante	<i>Foot_Right</i>	<i>Knee_Right</i>	Z
10. Pie izquierdo hacia delante	<i>Foot_Left</i>	<i>Knee_Left</i>	Z

Con este modo, un usuario sentado es capaz de utilizar cualquiera de los modos y movimientos disponibles de igual forma que un usuario que se disponga a utilizar Mokey de pie: doce movimientos predefinidos más cuatro posibles movimientos grabados.

3.5. Grabación de movimientos

En este apartado se explica el modo de grabación de movimientos alojado en la pestaña mostrada en la figura 3.13 de la ventana principal de Mokey. Esta pestaña consta de cuatro módulos idénticos al mostrado en la figura 3.23.

Un usuario que usa el programa puede no considerar suficientes, o acorde a sus necesidades, los doce movimientos predefinidos. Por ello, se propone el módulo de grabación de movimientos

personalizados. Con esta característica, un usuario puede personalizar hasta cuatro nuevos movimientos realizando una grabación con alguno de los cuatro huesos siguientes: *Hand_Right*, *Hand_Left*, *Foot_Right* o *Foot_Left*.

The image shows a software interface titled "Movement recorded 1". On the left, there is a checkbox labeled "Activation". To its right is a rounded rectangle labeled "Limb movement" containing four radio button options: "Right Arm", "Left Arm", "Right Leg", and "Left Leg". Further right is a "Key" dropdown menu with a downward arrow. To the right of the key is a "Delay" slider with a small square marker. At the bottom right of the interface is a button labeled "Record!".

Figura 3.23: Módulo movimiento grabado de la ventana principal de Mokey

Con cada uno de estos módulos es posible configurar un nuevo movimiento grabado por el usuario bajo sus propias preferencias y personalización. Mokey permite grabar hasta cuatro movimientos y amplía de esta forma las distintas posibilidades que se ofrecen al usuario, pudiendo este realizar un movimiento personalizado que no se encuentre entre los doce predefinidos.

La configuración de dichos movimientos grabados requiere de una persona que asista al usuario de Mokey durante su configuración. Esto es debido a que para grabar un movimiento es necesario que el usuario reproduzca la posición final del recorrido del movimiento en el mismo momento en el que la persona ayudante pulsa con el ratón en el botón de "Record!" de la interfaz gráfica. En ese mismo instante, el hilo seleccionado almacena la diferencia de las 3 coordenadas (X, Y, Z) de la extremidad seleccionada en el cuadro "Limb movement" respecto al punto de referencia como valor a alcanzar o superar para que Mokey ejecute el evento de teclado correspondiente. El hilo almacena en dos variables diferentes para cada eje el módulo y el signo de dicha diferencia para que posteriormente se puedan comparar las dos variables de forma independiente. Esto proporciona mayor exactitud y restricción en el cumplimiento de las condiciones de cada hilo, necesario debido a las múltiples posibilidades de grabación.

Dependiendo de la situación del usuario respecto al punto central de la imagen captada por Kinect, punto (0,0,0), las coordenadas captadas pueden ser negativas o positivas. Para la ejecución del evento de teclado relativo a cada movimiento es necesario que el módulo, además del signo, de la detección temporal del punto seleccionado sea igual o superior al valor de la diferencia almacenada al grabar el movimiento de forma simultánea en los 3 ejes de coordenadas.

En función del hueso seleccionado en el cuadro "Limb movement", el sistema coge como referencia en la comparación de movimientos puntos diferentes del esqueleto de Kinect. Esta distinción entre puntos de referencia está pensada para optimizar la grabación y obtener valores más estables en función del hueso seleccionado:

- **Right Arm:** Talón derecho (Punto *Ankle_Right*)
- **Left Arm:** Talón izquierdo (Punto *Ankle_Left*)
- **Right Leg:** Punto *Spine*
- **Left Leg:** Punto *Spine*

En este caso, los movimientos deben ser mucho más precisos para que sean reconocidos como los movimientos grabados previamente y para que Mokey lance el evento de tecla asociado al

mismo. Esto es debido a que es necesario cumplir la condición del movimiento en los tres ejes de coordenadas cartesianas que capta Mokey (X, Y y Z) a diferencia de los movimientos predefinidos que tan solo se evalúan en un único eje de coordenadas.

Para cada uno de estos movimientos grabados es posible asociar, al igual que al resto de movimientos, una tecla y un valor de *delay* para establecer la frecuencia de eventos que son simulados por Mokey al detectar el movimiento grabado.

3.6. Registro de datos

Uno de los usos para los que se ha diseñado Mokey es el hecho de que un especialista o asistente lo pueda emplear como ayuda en ámbitos terapéuticos. Para ello, es necesario tener una herramienta con el objetivo de recoger datos de uso y posibilitar de esta forma la toma de parámetros generales de usuario con el fin de poder realizar gráficos, informes y reportes detallados.

El funcionamiento de esta característica es simple. Una vez que se inicia el programa con la sesión de un usuario concreto, sea nuevo o no, y se configuran todos los parámetros de configuración requeridos para cada movimiento, este está listo para el registro de datos de uso. Una vez que el usuario pulsa el botón “Play!” de la ventana principal, el registro comienza y no se detiene hasta parar la ejecución del programa.

Mokey, una vez se comienza a ejecutar en segundo plano tras pulsar el botón de iniciar, crea un fichero con formato *csv* en la carpeta “registros” del directorio de la sesión abierta. Este fichero se crea con la siguiente nomenclatura:

DD MM - HH MM.csv

Donde DD, MM, HH y MM son el día, el mes, la hora en formato 24H y los minutos del día actual respectivamente. Se ha seleccionado este formato de fichero debido a la facilidad de gestión y edición con Microsoft Excel para su posterior creación de gráficos y estadísticas.

El contenido del fichero de registros creado por el programa de forma automática se puede observar en la tabla 3.3. En esta tabla se describe con detalle el contenido de cada una de las filas y columnas del fichero resultante.

En primer lugar, la primera fila hace referencia a la aplicación que se está ejecutando en primer plano y sobre la cual Mokey está simulando eventos de teclado. En segundo lugar, la siguiente fila relevante muestra la hora en la que se inicia el registro de datos, campo que se registra según la hora real del sistema operativo. A continuación, se encuentran dos filas indicando cada uno de los doce movimientos predefinidos configurables junto con el valor de desplazamiento o *displacement* configurado en la interfaz gráfica para cada movimiento, esté seleccionado como activo o no. Cada movimiento se registra en una columna diferente. El siguiente módulo es la parte principal del archivo de registro ya que se muestra, para cada movimiento, el registro de la magnitud del mismo cada 0,5 segundos. Este tiempo de medio segundo ha sido elegido debido a que presenta un equilibrio entre un tiempo demasiado corto, y que requiera demasiada carga computacional con un impacto en el rendimiento del equipo, y entre un espacio de tiempo que no sea suficiente como para registrar los movimientos realizados por el usuario. En el capítulo 4 se realizan pruebas sobre dicho rendimiento. Esta magnitud registrada para cada movimiento hace referencia al resultado de cada una de las condiciones de los algoritmos de cada uno. Por ejemplo, en el caso del movimiento de la pierna hacia delante, como se ha descrito en el

apartado 3.4, se registra la diferencia entre los puntos del esqueleto del pie y de la cadera. Al final del fichero se guarda la fecha y hora del fin del registro y el nombre del archivo de configuración seleccionado al cargar el programa.

Tabla 3.3: Contenido fichero csv de registros de parámetros de uso

	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7	Col. 8	Col. 9	Col. 10	Col. 11	Col. 12	Col. 13
Fila 1	Aplicación ejecutada												
Fila 2	vacío												
Fila 3	Fecha y hora del inicio del registro. Formato "dd/mm/aaaa hh:mm:ss"												
Fila 4	vacío												
Fila 5	Movimiento	1	2	3	4	5	6	7	8	9	10	11	12
Fila 6	Umbral(m)	Valores parámetro displacement configurado para cada movimiento											
Fila 7	vacío												
Fila 8	Tiempo(s)	Medida(m)											
Fila 9	0,0	Registro temporal de la magnitud de cada movimiento											
Fila 10	0,5												
Fila 11	1,0												
...	...												
Fila N	$A_{n-1} + 0.5$												
Fila N + 1	Vacío												
Fila N + 2	Fecha y hora del fin del registro. Formato "dd/mm/aaaa hh:mm:ss"												
Fila N + 3	Vacío												
Fila N + 4	Vacío												
Fila N + 5	Nombre archivo de configuración cargado al iniciar Mokey												

Con todos los datos registrados en el archivo .csv guardado, el terapeuta puede elaborar un tratamiento del fichero para mostrar datos relevantes, gráficos o tablas a partir de todo el registro realizado. Debido a que en el documento se registran los valores de desplazamiento de los miembros corporales en función del tiempo, es posible realizar gráficos para analizar cuándo se cumplen las condiciones de los movimientos y se lanzan eventos de teclado. También pueden ser empleados para la supervisión de determinados ejercicios por parte de los usuarios, detectando la amplitud concreta de los movimientos de articulaciones o extremidades.

Capítulo 4. RESULTADOS Y ANÁLISIS

A lo largo de este capítulo, se elabora un análisis de los resultados de las pruebas realizadas a Mokey para comprobar su efectividad y funcionamiento bajo casos reales tanto con personas con discapacidad y capacidad motora reducida como con personas sanas. Estas pruebas se realizaron atendiendo al aspecto físico del usuario y su experiencia final: ejecución, configuración, experiencia de juego, etc. Como evaluación dentro del marco de aplicaciones actuales, se compara con la aplicación que más se asemeja en objetivos y funciones, con FFAST.

Para este proyecto se han realizado pruebas de rendimiento, ampliamente analizadas en el texto de Marcos López en [1]. También se ha realizado en ese análisis una comparación a nivel de rendimiento y utilización de recursos con FFAST. Estas pruebas de rendimiento se han basado en el análisis y la comparación en el uso de recursos de CPU de los procesos relativos a Mokey y a FFAST, así como en el consumo de memoria RAM que requieren ambos programas, tanto en estado de reposo como en ejecución junto con la aplicación sobre la cual se envían los eventos de teclado.

Todas las pruebas se han realizado ejecutando el programa en un Intel (R) Core 2 6400 con 2.13 GHz y 4 Gb de memoria RAM.

4.1. Experiencia configuración interfaz gráfica

En un primer lugar, la prueba inicial realizada con Mokey es la referente al tiempo de hacer una configuración concreta de acuerdo a las necesidades del usuario. Las pruebas se han realizado en el centro CITSEM con usuarios con conocimiento previos de los conceptos bajo los que se basa Mokey y usuarios con ninguna experiencia previa con la aplicación. Para comparar la experiencia en el proceso de configuración se compara este proceso con FFAST con el objetivo de establecer una comparación de ambos programas a la hora de la facilidad de configuración.

Para la realización de la prueba referente al tiempo de configuración del programa, se han seleccionado dos configuraciones concretas para utilizar de forma simultánea:

- Brazo derecho a la derecha y pulsación tecla letra “A”
- Brazo izquierdo a la izquierda y pulsación tecla letra “B”

La configuración de estos dos movimientos es simple, sin necesidad de requerir de los movimientos de grabación. Se ha seleccionado así para evaluar la experiencia de usuario en cuanto a si la interfaz diseñada es intuitiva o no.

La prueba se ha realizado a dos personas diferentes, dos estudiantes. El primer usuario realiza la configuración de los dos movimientos descritos con conocimiento previo tanto de Mokey, como de FFAST. Sin embargo, el segundo usuario de la prueba se enfrenta a ambas interfaces gráficas sin conocimiento previo, tan solo con una breve explicación.

En lo relativo a Mokey, los dos usuarios se encuentran con la Kinect conectada y el programa abierto en la ventana principal, con la sesión iniciada y con la interfaz de configuración con todos los valores de los parámetros por defecto y todos los movimientos desactivados. Ambos usuarios encuentran intuitivamente los controles relativos a las dos configuraciones requeridas. El usuario uno tarda tan solo 22 segundos en configurar los dos movimientos mientras que el segundo usuario, sin experiencia previa en Mokey, realiza las dos configuraciones en 72 segundos.

Para que estas mediciones de tiempo cobren un valor real es necesario realizar una comparación con aplicaciones existentes similares. Por ello, se someten estos mismos usuarios a la configuración de los mismos parámetros, pero en el programa FFAST.

4.1.1. Comparación con FFAST

Para establecer una medición real, se selecciona FFAST para realizar la misma prueba que con Mokey con los mismos usuarios. Para ello, una vez se miden los tiempos de configuración de los movimientos requeridos de los dos usuarios configurando Mokey, se someten a la misma configuración en FFAST. Para ello, para estar en igualdad de condiciones con la prueba realizada con Mokey, se presenta la aplicación en el menú principal de configuración y la cámara Kinect conectada y listo para usar. El resultado de la prueba se muestra en la tabla 4.1. El tiempo mostrado en la tabla hace referencia desde que se entrega la ventana de configuración con los valores por defecto hasta que se ejecuta la aplicación con el programa sobre el que se envían los eventos del teclado.

Tabla 4.1: Resultados tiempos de configuración para Mokey y FFAST

	Mokey	FFAST
Usuario 1: con experiencia previa	22 segundos	156 segundos *
Usuario 2: sin experiencia previa	72 segundos	240 segundos **

Observaciones prueba:

* De los 156 segundos resultado de la medición, aproximadamente 36 segundos han sido destinados a la conexión de Kinect con FFAST. Proceso que el programa realiza siempre tras realizar la configuración.

** Tras 4 minutos, el usuario 2 no ha sido capaz de acabar de configurar las directrices requeridas. Tras observar que no sabía proceder, se proporcionaron unas pequeñas pautas que, sin embargo, no sirvieron para conseguir configurar FFAST como se requería.

Como se puede observar en la tabla 4.1, tener un conocimiento previo de los conceptos básicos de ambos programas es un aspecto relevante a la hora de configurar ambos programas. Sin embargo, este hecho se acentúa a la hora de configurar FFAST debido a que el segundo usuario no ha sido capaz de configurar el programa sin la ayuda de los supervisores. Con Mokey, la diferencia entre tener un conocimiento general del programa y no tenerlo, hace que se tenga que dedicar cerca de 40 segundos más a la configuración del mismo. Sin embargo, con FFAST este dato aumenta su tiempo considerablemente debido a que, sin conocimiento previo o pequeñas pautas, se hace inviable la configuración.

De forma general, atendiendo al tiempo de configuración entre los dos programas que dedica la misma persona en su configuración, también se pueden obtener datos relevantes y diferencias entre ambos programas. Analizando los datos mostrados en la tabla 4.1, una misma persona necesita dedicar más tiempo a la configuración de los mismos movimientos en los dos programas. Mientras que el usuario 1 le dedica 22 segundos a la configuración de Mokey, este mismo usuario emplea 156 segundos en configurar FFAST, lo que supone un aumento de cerca de un 663% de tiempo respecto al tiempo dedicado en Mokey.

No menos importante también, es el hecho del tiempo que tarda cada uno de los dos programas en ejecutarse desde que el usuario configura los movimientos requeridos y pulsa el botón correspondiente para comenzar. En Mokey, una vez el usuario pulsa el botón “Play!”, el proceso

principal crea los hilos necesarios para cada movimiento configurado y se ejecuta bajo la configuración establecida de forma prácticamente instantánea. Sin embargo, en FFAST, este mismo proceso se alarga a tiempos cercanos a los 30 o 40 segundos, lo que hace una experiencia de uso más lenta.

La interfaz gráfica de Mokey es simple e intuitiva de forma que hace que los usuarios sean capaces de configurar la Interfaz Natural satisfactoriamente en un tiempo asumible de menos de minuto y medio en la mayoría de casos, tenga conocimiento previo el usuario o no. Sin embargo, en el caso de FFAST, al tratarse de una interfaz con un aspecto mucho más personalizable y con multitud de opciones y parámetros de regulación, a la hora de la configuración de un usuario sin conocimiento, se hace un proceso difícil, y muchas veces inviable, elevando el tiempo de este proceso a más de los cuatro minutos. Con FFAST, un usuario con conocimiento previo de la herramienta completa su configuración sin problemas, pero, aun así, en un tiempo mayor que lo que tarda una persona sin experiencia previa en completar la configuración en Mokey.

4.2. Interacción de Mokey con aplicaciones de terceros

El propósito para el que ha sido diseñado Mokey es para el de ser un *Middleware* entre el usuario y la aplicación en cuestión que se quiere controlar. Por ello, es importante analizar la compatibilidad entre Mokey y estas aplicaciones de terceros. En [1] se detalla esta prueba, pero atendiendo al impacto que estas aplicaciones tienen en el uso de Mokey en cuanto a memoria RAM y uso de CPU se refiere, así como en las limitaciones en los algoritmos implementados.

Para realizar esta prueba se ha hecho una selección de aplicaciones y programas de uso común por la mayoría de la población de forma frecuente. El resultado se detalla a continuación.

- **Google Chrome:** Permite la navegación de páginas web simples siempre y cuando estas puedan ser controladas mediante eventos de teclado. Debido a que, mediante Mokey, no está permitido escribir direcciones URL a causa de permitir tan solo la configuración de 16 teclas de forma simultánea, las páginas deben estar almacenadas en los marcadores o favoritos del navegador.
- **Aplicaciones bajo Adobe Flash en navegador web:** Gracias a la optimización de Mokey, es posible controlar aplicaciones ejecutadas bajo Adobe Flash en cualquier navegador, teniendo en cuenta las limitaciones de Mokey. No es posible escribir cadenas de texto, pero sí simular los controles básicos para el control de multitud de aplicaciones y juegos. Mokey actúa sobre la ventana con el contenido Flash como si de la misma ventana del navegador se tratase.
- **Avast:** Es un ejemplo de un programa de tipo antivirus que permite su uso sin necesidad del ratón, convirtiéndolo en totalmente compatible con Mokey. Es posible programar un análisis del equipo, control de los escudos activos, entre otras tareas.
- **Microsoft Word:** Mokey no permite escribir un documento íntegro, pero sí que permite navegar por un documento existente a modo de lectura y realizar una edición limitada. Mokey presenta una ventaja a personas discapacitadas en este aspecto debido a que permite la lectura de documentos privados sin necesidad de asistencia por parte de otra persona.

- **Microsoft PowerPoint:** Al igual que con Microsoft Word, Mokey no permite la redacción y edición completa de presentaciones. Lo que sí permite es realizar presentaciones de diapositivas sin necesidad de ningún dispositivo remoto, con el movimiento, de forma que una persona con discapacidad pueda hacer una presentación al público con total autonomía.
- **IceCream Ebook Reader:** Es un lector de libros electrónicos que permite varios formatos. Con Mokey es posible navegar por cualquier libro digital, incluso estableciendo marcadores, permitiendo la lectura completamente autónoma a usuarios con movilidad reducida.
- **Tetris:** Es el juego por referencia utilizado en Mokey para la realización de pruebas. El popular juego es totalmente compatible con Mokey, permitiendo su control con multitud de combinaciones de movimientos debido a que el juego requiere tan solo de cuatro teclas para su completo control.
- **Skype:** Con Mokey no es posible mantener conversaciones de mensajes de texto, pero sí que permite establecer llamadas y videollamadas con los contactos almacenados en la libreta de direcciones. Una opción muy interesante para personas con movilidad reducida debido a que permite su comunicación sin necesidad de ayuda.
- **League Of Legends:** Debido a que es un videojuego que requiere gran parte de los recursos del ordenador, Mokey no puede ser empleado para su control en cualquier ordenador. Además, al ser un videojuego con dependencia del ratón, no es posible su control de forma totalmente autónoma.
- **Minecraft:** Debido a la gran versatilidad en la configuración, con Mokey es posible controlar gran parte de las acciones de este famoso videojuego de bloques permitiendo de esta forma su utilización a personas con movilidad reducida. El único problema es que la cámara depende de la rotación del ratón. Con Minecraft es posible diseñar aplicaciones específicas para la realización de determinados ejercicios o rutinas de rehabilitación.

4.3. Pruebas con voluntarios

Para la evaluación de Mokey en personas y comprobar su efectividad real, se han hecho pruebas con dos grupos de personas diferentes: primero con personas con algún tipo de discapacidad y a continuación con personas sin discapacidad alguna. Siendo, en cada uno de los grupos, personas de diferente edad para lograr una muestra más amplia.

4.3.1. Pruebas con personas con discapacidad

Gracias a ASEM (*Asociación de Enfermedades Neuromusculares de Madrid*) y a la Fundación TAMBIÉN (asociación que promueve las actividades deportivas para personas con discapacidad física), se presentan como voluntarias 11 personas en total, 3 adultos y 8 niños, para probar Mokey y su efectividad.

Los 11 voluntarios se presentan con diferentes tipos de discapacidad, tal y como se detalla en la tabla 4.2. Cada una de las personas voluntarias se somete a un protocolo definido de pruebas con el fin de evaluar si Mokey cumple o no las necesidades de cada usuario en particular, especialmente atendiendo a la discapacidad y sus dificultades de movimiento.

También es importante recoger las posibles necesidades de cada uno de los tipos de discapacidad para, si presta dificultades en la versión actual, actuar de forma efectiva como trabajo futuro en las futuras versiones de Mokey de forma que se cubran las demandas de cada persona para el uso de aplicaciones de uso general.

Tabla 4.2: Voluntarios para las pruebas de Mokey

Voluntario	Discapacidad	¿Silla ruedas?	Género	Age
1	SMA (Atrofia muscular espinal) tipo 2	Si	Hombre	5
2	SMA (Atrofia muscular espinal) tipo 2	Si	Hombre	13
3	CP (Parálisis Cerebral)	Si	Hombre	11
4	CP (Parálisis Cerebral)	Si	Hombre	12
5	Hipotonía	Si	Hombre	12
6	BMD (Distrofia Muscular de Becker)	No	Hombre	13
7	DMD (Distrofia Muscular de Duchenne)	Si	Hombre	15
8	DMD (Distrofia Muscular de Duchenne)	Si	Hombre	16
9	FSH (Distrofia Muscular Facioescapulohumeral)	Si	Mujer	43
10	FSH (Distrofia Muscular Facioescapulohumeral)	No	Mujer	49
11	PPS (Síndrome Post-Polio)	No	Mujer	50

El protocolo relativo a las pruebas que sigue cada una de las personas voluntarias consiste en la evaluación de la experiencia de usuario en el uso de 4 programas de uso cotidiano: **Microsoft PowerPoint**, un **lector de libros electrónicos**, **Skype** y, por último, el popular juego **Tetris**. La razón por la cual se han seleccionado estos programas es debido a la utilidad que pueden aportar al colectivo de personas con discapacidad. Todo ello siguiendo siempre ese mismo orden.

Antes de comenzar las pruebas, se pregunta a cada uno de los usuarios qué tipo de discapacidad tiene y qué limitaciones de movimiento conlleva. Tras ello se decide qué movimientos se ajustan más a las necesidades de cada usuario para, a continuación, realizar una breve explicación del funcionamiento de Mokey y de cada uno de los movimientos que es posible configurar en el programa, incluyendo la posibilidad de grabar sus propios movimientos personalizados. Esta explicación se realiza con la ayuda de una presentación de PowerPoint que el propio usuario puede controlar mediante movimientos. Tras la breve explicación, el usuario comienza a utilizar las aplicaciones elegidas para las pruebas ajustando los parámetros de cada movimiento elegido a las necesidades de cada uno. Si durante el uso de los programas, el usuario no se encuentra cómodo, se rectifican los parámetros de cada movimiento seleccionado en tiempo real. Si el usuario muestra signos de fatiga o cansancio, se para a descansar durante el periodo de tiempo necesario. Al finalizar el proceso, todos los participantes rellenan un formulario contando la experiencia con cada uno de los programas.

Una vez finalizadas las pruebas, se elabora una síntesis de los resultados obtenidos en las mismas con el fin de evaluar la aplicación. En la tabla 4.3 se anotan los resultados estadísticos de los movimientos configurados más usados y con mayor tasa de éxito de entre todas las configuraciones de los usuarios.

Además, de los resultados de la tabla 4.3 y de las anotaciones y registros obtenidos durante las pruebas, se puede deducir que:

- Siempre que el usuario pueda realizar un movimiento, la tasa de éxito del mismo es mayor al 80%.
- Para muchos usuarios, los valores por defecto de los parámetros de *delay* y *displacement* han resultado válidos.
- Tan solo una persona prueba los movimientos relativos a los pies.
- Nadie muestra interés en probar alguno de los 4 movimientos personalizables mediante la grabación.
- Para los usuarios en silla de ruedas, aun con el modo *Wheelchair* activado, se han encontrado dificultades para detectar los movimientos relativos a los brazos. Pese a esto, se han encontrado siempre alternativas con el uso de otros movimientos diferentes.
- Los movimientos relativos a la inclinación del tronco son los que mayor tasa de éxito han dado, con valores cercanos al 100%. Además, los usuarios han manifestado sentirse cómodos con ellos.

Tabla 4.3: Resultados estadísticos del uso de movimientos

Movimiento	Veces probado	Veces ajustado	No funcionó	Tasa de éxito
Brazo Derecho arriba	26	2	1	96
Brazo Izquierdo arriba	26	2	1	96
Brazo Derecho derecha	11	0	2	82
Brazo Izquierdo Izquierda	11	0	1	91
Tronco Derecha	12	3	0	100
Tronco Izquierda	12	3	0	100
Pie Derecho arriba	1	1	0	100
Pie Izquierdo arriba	1	1	0	100

De los cuatro programas probados, tan solo el Tetris ha resultado problemático para los usuarios, teniendo que ajustar en repetidas ocasiones los movimientos configurados para cada caso para conseguir sentirse cómodos al utilizarlo. En los otros tres programas, los usuarios se han sentido bien desde el primer momento, sin ser necesario un proceso previo de adaptación. En el caso del Tetris, al tratarse de un juego que requiere más exactitud y precisión en los movimientos, ha sido más problemático llegar a la configuración óptima, necesitando mayor periodo de adaptación. En la tabla 4.4 se muestra el número de veces que se ha probado cada programa y cuántos intentos fueron necesarios para que funcionen correctamente, habiendo periodo de adaptación entre intento e intento. Este periodo de adaptación consiste en la prueba libre de los movimientos elegidos durante un corto periodo de alrededor de 10-20 segundos mientras se explica el funcionamiento del programa en cuestión.

Tabla 4.4: Número de intentos para cada programa

Programa	Veces probado	Funciono a la 1ª	Funciono a la 2ª	No funcionó
PowerPoint	10	9	1	0
Lector Ebooks	9	9	0	0
Skype	9	9	0	0
Tetris	9	2	6	1

Como se puede observar, la tasa de éxito tras el periodo de adaptación ha sido del 100% salvo en el caso del Tetris, que se sitúa en valores próximos al 88%, un 22% tras el primer intento. Tan solo un usuario no ha podido adaptarse al juego del Tetris a causa de la imposibilidad de Mokey de proporcionar movimientos que se ajusten a sus necesidades. En este caso, el usuario en cuestión, permanecía en silla de ruedas con la única movilidad del tronco y los brazos de forma reducida. Debido a que para jugar al Tetris son necesarios 4 controles, no fueron suficientes los 2 movimientos relativos al tronco ya que la escasa movilidad de los brazos y pies no producía el efecto deseado, ni siquiera con los movimientos grabados. En el caso del uso de PowerPoint, solo hay un caso de un usuario que necesitara una adaptación y un segundo intento, en este caso, se trata de una persona de corta edad al que el programa no suscitaba interés. No todos los voluntarios siguieron el protocolo al completo debido al cansancio, falta de atención o falta de interés debido de la edad.

De todas las pruebas con voluntarios se han recogido ficheros de datos que, por privacidad y protección de datos de los voluntarios, se han marcado como confidenciales. Por ello, se presentan las observaciones resultantes de dichas pruebas. Tras la anotación de las observaciones, se llega a la conclusión de que es necesario, en términos generales, un periodo de adaptación para aplicaciones que requieran de interacción rápida y razonada del usuario, como es en el caso de los juegos como el Tetris. En el caso de aplicaciones simples que requieran pocos movimientos no hay a priori mayor complicación. Como observaciones de las pruebas, se puede deducir lo siguiente:

- **El uso de silla de ruedas conlleva problemas para el uso de Mokey.** En ocasiones se observan problemas en la detección de las coordenadas de los puntos relativos a los brazos en usuarios de sillas motorizadas con mando en uno de los brazos de la silla. Dicho mando provoca falsas detecciones en el esqueleto de Kinect. Esto tan solo afecta a los movimientos de brazos arriba, no en los relativos al desplazamiento lateral de los brazos. Pese a este problema, el uso de silla de ruedas no es un impedimento para el uso del programa, todos los usuarios en silla de ruedas han hecho uso del mismo.
- **El retardo máximo entre eventos de teclado, en ocasiones, es insuficiente.** Debido al tipo y grado de discapacidad de algún usuario, el retardo máximo es insuficiente para evitar enviar dos eventos consecutivos para la misma tecla. Este hecho provoca frustración en los usuarios.
- **Resultado positivo en usuarios que no podían repetir el mismo movimiento a causa de su discapacidad.** Las personas con problemas psíquicos tienen problemas para, conscientemente, repetir el mismo movimiento de forma consecutiva. Pese a ello, el programa muestra un punto a favor en la detección de forma general y simple de cada movimiento. Por ejemplo, usuarios con Parálisis Cerebral han tenido una experiencia muy positiva debido a que no es necesario repetir exactamente un movimiento, basta con superar un umbral en una determinada dirección dependiendo del movimiento.
- **Los movimientos son insuficientes en algunos casos de discapacidad.** En el tránsito de las pruebas, una persona no obtuvo suficientes movimientos adaptables a sus necesidades para manejar el juego de Tetris. En concreto esta persona tenía que permanecer en silla de ruedas pudiendo mover tan solo el tronco y los brazos de forma reducida y limitada. Al precisar el juego de 4 movimientos diferentes para su control, tan solo pudo adaptarse a los 2 movimientos de tronco. Todas las configuraciones posibles con los brazos, incluyendo con el modo de grabación, no fueron suficiente para

conseguir una configuración estable a causa de la poca movilidad de las extremidades. Debido a ello, los movimientos predefinidos no son suficientes para satisfacer las necesidades del juego. Se necesitan más movimientos de brazos y tronco para abrir el abanico de posibilidades de configuración, incluso, con gestos de las manos o faciales.

- **Amplitud insuficiente en casos de Distrofia Muscular.** Al igual que en el caso anterior de los movimientos, en ocasiones, debido al tipo de discapacidad, la precisión de Kinect no es suficiente como para detectar movimientos en los usuarios. Personas con poca movilidad en extremidades, cabeza o tronco, pueden no encontrar suficiente precisión en la detección de los movimientos por parte de los algoritmos de Mokey debido al rango predefinido del parámetro *displacement* configurable en cada movimiento.
- **La grabación de movimientos con usuarios en silla de ruedas es errática.** Debido a las interferencias que causa la silla en sí, y el mando en el caso de las sillas motorizadas, estos usuarios tienen dificultades para que el programa almacene correctamente el movimiento que se quiere grabar. Estas interferencias inducen en falsas detecciones en los puntos de la mano e imposibilita su correcta detección. Además, una persona en silla de ruedas no suele poder mover las piernas, por lo que tampoco le valdría grabar los movimientos con los pies. Una posible solución de este problema sería la de añadir más puntos diferentes para realizar la grabación –como por ejemplo cabeza o tronco–, gestionar de forma eficiente las falsas detecciones mediante el código fuente del programa o, incluso, añadir un modo sentado de grabación en el que cambien los puntos de referencia de los algoritmos al igual que en los 12 movimientos predefinidos.
- **La respuesta de Mokey en detección de movimientos en distancias cortas es muy positiva.** A distancias menores de 1,5 metros, Kinect se ha comportado de forma satisfactoria en la mayoría de casos. Una vez se detecte el esqueleto del usuario de Mokey, este se puede acercar hasta 1 metro de cercanía sin problemas, incluso con los movimientos del tronco.
- **Para los movimientos relativos al tronco, es necesario mayor estabilidad del cuerpo.** En el caso de tener dificultad de mantener una posición erguida, la detección de estos movimientos no funciona correctamente en caso de tener configurados también movimientos en los brazos. Este problema es debido a las interferencias que suponen los movimientos de los brazos en los movimientos del tronco a causa de que los algoritmos de los primeros emplean la misma referencia que los algoritmos de los segundos: punto *Spine*. Al utilizar la misma referencia, y no poder mantener el cuerpo erguido, al intentar hacer un movimiento con los brazos, puede cumplirse también las condiciones del movimiento del tronco, lo cual lleva a interferencias y confusión en el programa respecto al movimiento realizado. No se observan problemas si tan solo se configuran los movimientos relativos al tronco.
- **Un usuario sugirió la colocación de Kinect detrás del cuerpo.** Con ello, se podría obtener mayor precisión y acercamiento a la pantalla del ordenador. Sin embargo, se pierden algunos movimientos relativos al tronco y pies.

4.3.2. Pruebas con personas sin discapacidad

En una segunda jornada de pruebas, se realizaron pruebas con Mokey a un grupo de 8 niños entre 7 y 13 años. Del mismo modo que en las pruebas con personas con alguna discapacidad,

cada persona voluntaria se somete al protocolo de pruebas destinado a evaluar la funcionalidad completa real del programa. Para estas pruebas ha sido seleccionado sólo el Tetris debido a la simplicidad de juego y su uso extendido en la sociedad. La edad de los participantes también es determinante en la elección de la aplicación sobre la que realizan las pruebas ya que, al ser usuarios de menos de 13 años, es importante que mantengan la atención el mayor tiempo posible.

En esta jornada de pruebas el protocolo ha consistido en una breve explicación inicial del programa y su funcionamiento para, a continuación, proceder a acordar los movimientos empleados para jugar. Posteriormente, los usuarios han tenido un periodo de tiempo para familiarizarse con los movimientos y su funcionalidad. Tras ese periodo, el usuario inició una partida y se observó el resultado. Si el usuario lo requería, era posible iniciar una nueva partida.

Tras observar el comportamiento de los 8 usuarios que han manejado el Tetris, se recogen los resultados del uso de cada movimiento probado y se elabora la tabla 4.5. En ella se puede observar la cantidad de veces que ha sido seleccionado cada movimiento, cuántas veces ha requerido algún ajuste adicional o si no ha funcionado. Ningún usuario consiguió utilizarlo correctamente sin fallos a la primera, siendo indispensable el periodo de adaptación previo. Tras un primer análisis, todos los usuarios han tenido una tasa de éxito del 100%, tras el periodo de adaptación, en todos los movimientos seleccionados.

Tabla 4.5: Movimientos probados por personas sin discapacidad y tasa de éxito

Movimiento	Veces probado	Veces ajustado	No funcionó	Tasa de éxito
Brazo Derecho arriba	6	0	0	100
Brazo Izquierdo arriba	6	0	0	100
Brazo Derecho derecha	1	0	0	100
Brazo Izquierdo Izquierda	1	0	0	100
Tronco Derecha	1	0	0	100
Tronco Izquierda	1	0	0	100
Pie Derecho arriba	6	0	0	100
Pie Izquierdo arriba	6	0	0	100

Debido a la corta edad de los voluntarios más pequeños, no ha sido posible probar Mokey con el Tetris con todos ellos, a causa de que no conocían el juego. Aun así, sí que se ha examinado el comportamiento de cada uno y la respuesta a los movimientos que querían hacer proponiendo en tiempo real jugar con las piezas de Tetris, su rotación, colocación, etc. de una forma animada y entretenida. Pese a la satisfactoria respuesta de los voluntarios a los movimientos, la experiencia global no ha sido positiva debido al desconocimiento del juego. Como observaciones de las pruebas se detalla lo siguiente:

- **La elección del programa para realizar la prueba no ha sido la correcta.** La edad de los voluntarios ha sido inferior a lo esperado y gran parte de ellos no conocían el juego del Tetris. En concreto, los niños más pequeños, no entendían el funcionamiento del juego y la realización de la prueba ha supuesto un esfuerzo mayor al esperado. La experiencia general de los voluntarios no ha sido positiva debido a que, a la vez que comprendían el funcionamiento de Mokey, debían comprender el funcionamiento del Tetris. La solución hubiese sido hacer la prueba con un juego acorde al rango de edad.

- Un número significativo de usuarios presentan **dificultades en la comprensión de cómo gestiona los movimientos Mokey**. Muchos usuarios, al realizar un movimiento, dejaban la extremidad en cuestión en la posición de máxima amplitud y no comprendían que para hacer el siguiente movimiento debían volver al estado de reposo. En varias ocasiones, los voluntarios han levantado ambos brazos a la vez. Tras realizar indicaciones a estos usuarios, corregían de nuevo su posición, aunque volvían a caer en el error más tarde. Este hecho generaba frustración en el usuario ya que el juego no respondía como se requería, aunque la Interfaz Natural sí que reaccionase bien. Este hecho no se ha dado en casos de usuarios con discapacidad y ha llevado, en este caso, a una necesidad de un periodo de adaptación mayor.
- Muchos **movimientos sugeridos en primera instancia por los voluntarios eran referentes a partes del cuerpo no contempladas** en el programa. Algunos usuarios sugerían controlar el juego con movimientos de ojos, gestos con las manos únicamente o movimientos con la cabeza. Gran parte de los movimientos propuestos en primera instancia por los usuarios no están implementados debido a limitaciones de Kinect. Este aspecto puede ser una línea a tener en cuenta en un futuro con la nueva Kinect 2.0.

Durante la realización de las pruebas se observan las siguientes incidencias relacionadas en su mayoría con el procesado del ordenador elegido para las pruebas:

- **Falsas detecciones** debido al fondo captado por Kinect. Detección de otras personas u objetos del espacio de trabajo. Se produce 1 excepción controlada del programa debido a este hecho.
- **Problemas con el ordenador** debido al sobrecalentamiento y un estado de escritura en disco del 100% en periodos prolongados. Ralentización de los procesos debido a la antigüedad del ordenador empleado. Surgen 4 excepciones controladas del programa durante las pruebas debido a esto.
- **Problemas en la cola de procesos del ordenador** generando retraso en la ejecución de órdenes.
- Al probar **cambiar el tiempo de muestro del archivo de registro de 0,5 a 0,2 segundos**, el procesamiento de la CPU del ordenador no era suficiente para gestionar todos los datos.

4.4. Rendimiento registro de datos

Con el fin de comprobar si los resultados obtenidos en las pruebas con usuarios pueden ser válidos debido al bajo rendimiento del ordenador usado, se ha realizado un estudio comparativo con un ordenador de rendimiento alto, acortando en él el periodo entre muestras en la toma de datos. El objetivo es el de observar cómo afecta la capacidad de procesado del ordenador y el periodo de muestro a la hora de generar el fichero y, además, comprobar si el periodo de muestro de 0,5 segundos es suficiente para registrar la máxima amplitud de cada movimiento.

Los dos equipos usados fueron:

- Equipo 1 (pruebas usuarios): *Intel® Core 2 6400 a 2,13 GHz.i7-4790 a 3,6 GHz.*
- Equipo 2 (alto rendimiento): *Intel® i7-4790 a 3,6 GHz.*

Para la prueba, se han realizado 2 mediciones con una persona sana, registrando sus movimientos en los dos equipos al mismo tiempo. En la primera medición, ambos equipos registraron los datos cada 0,5 segundos; en la segunda medición, el equipo 2 registró a 0,2 segundos, manteniendo el equipo 1 igual. Cada una de las mediciones consistió en la repetición de 3 series de 5 movimientos laterales de cada brazo (movimientos 5 y 6). La primera serie contenía 5 movimientos lentos, la segunda 5 movimientos considerablemente más rápidos y, la tercera, 5 movimientos lo más rápido posible. Las 2 mediciones se han realizado ejecutando el juego de Tetris, comprobando visualmente que cada movimiento fue detectado correctamente como un evento, moviendo el bloque que caía en el juego.

En la figura 4.1 se puede observar el gráfico de la primera medición, con el equipo 1 a un tiempo de muestreo de 0,5 segundos. Las curvas muestran las distancias entre el punto de referencia *Spine* en el eje Y en relación al tiempo transcurrido en el eje X, en naranja para el brazo derecho y en verde para el brazo izquierdo. Se aprecian las 3 zonas relativas a cada una de las series de 5 movimientos con la progresiva velocidad en la ejecución. La amplitud de los movimientos realizados parte hacia valores próximos a 0,45 metros en toda la medición teniendo en cuenta que en la posición de reposo se parte con una diferencia estable de 0,2 metros aproximadamente. Se puede observar que, en la primera serie realizada a velocidad lenta, los 5 movimientos de cada brazo se muestran con claridad en la gráfica, lo que no ocurre en la segunda serie a mayor velocidad y, en menor medida, en la tercera. A medida que se aumenta la velocidad de los movimientos, el tiempo de muestreo del registro de 0,5 segundos no es suficiente y no capta la máxima amplitud de cada movimiento.

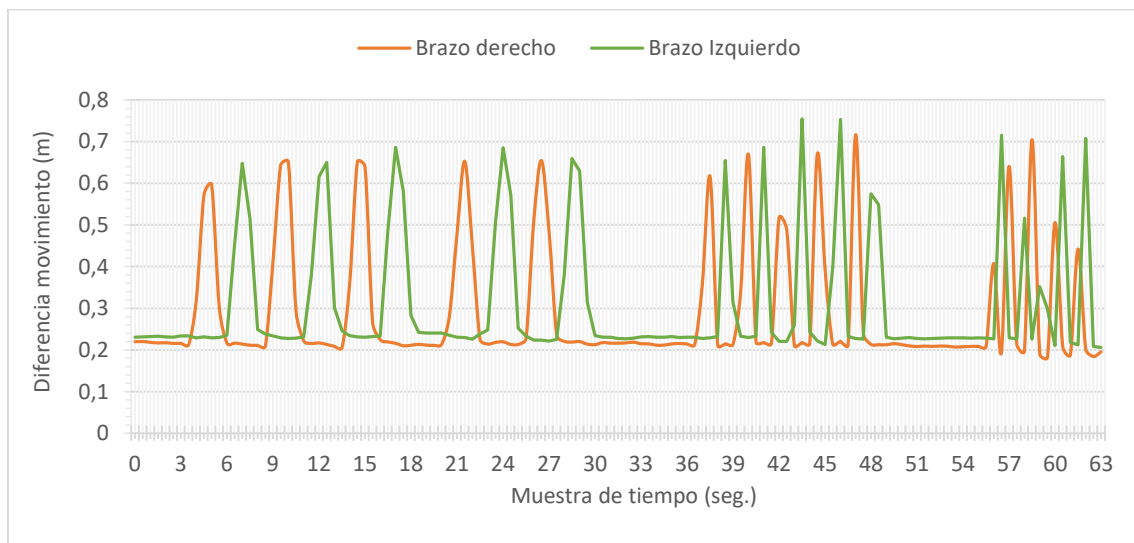


Figura 4.1: Gráfico del registro de datos con el equipo 1 y frecuencia de muestreo de 0,5 segundos

En la figura 4.2, se muestran los resultados de la misma prueba en el equipo más potente, el equipo 2, manteniendo el valor del tiempo de muestreo en 0,5 segundos. En gráfico se puede observar un resultado muy similar al de la figura 4.1. En la primera serie de movimientos lentos, la detección es correcta y se registran satisfactoriamente todos los movimientos realizados. Sin embargo, en la segunda serie realizada a una mayor velocidad, se puede apreciar que ciertos movimientos no han sido captados con su máxima amplitud, efecto que se hace más notorio en la tercera serie a velocidad rápida, donde no se registra correctamente la amplitud de todos los movimientos. Tras esta prueba se confirma que el procesador no es determinante a la hora de realizar el registro a un tiempo de muestreo de 0,5 segundos, ya que el resultado en las dos primeras pruebas es similar.

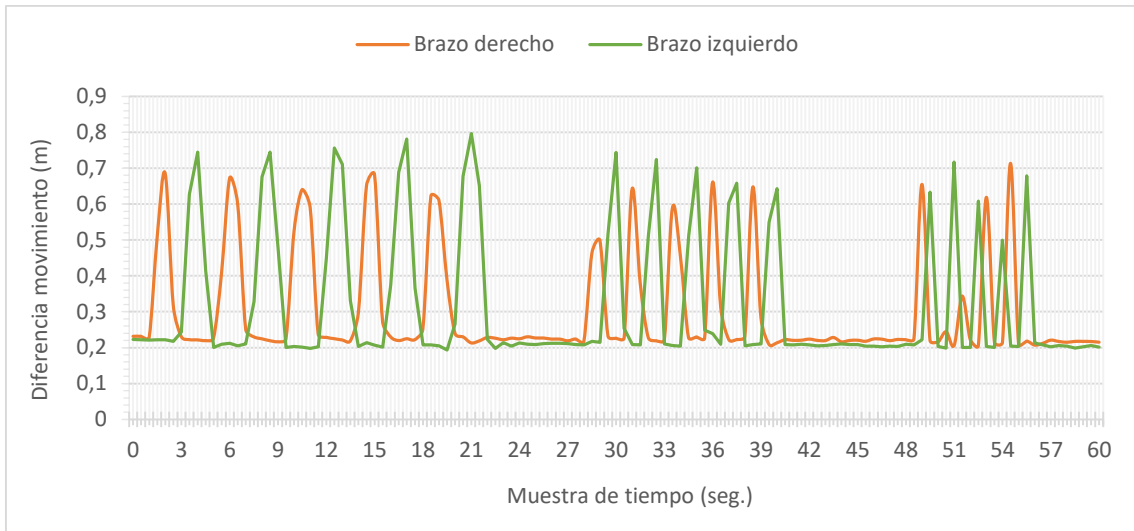


Figura 4.2: Gráfico del registro de datos con el equipo 2 y frecuencia de muestreo de 0,5 segundos

En la segunda medición se ha realizado la misma prueba en el equipo 2 cambiando el periodo de muestreo del registro de datos a 0,2 segundos. El resultado se puede observar en la gráfica de la figura 4.3. En esta ocasión, las 3 series a distintas velocidades se muestran correctamente tanto en número de movimientos como en la amplitud de los mismos, valor que alcanza los 0,45 metros de media. Comparando los resultados de esta medición con la anterior, realizada también en el equipo 2, se deduce la importancia del valor del periodo de muestreo al registrar los datos, ya que en esta tercera prueba se han registrado correctamente todos los movimientos realizados al usar Mokey.

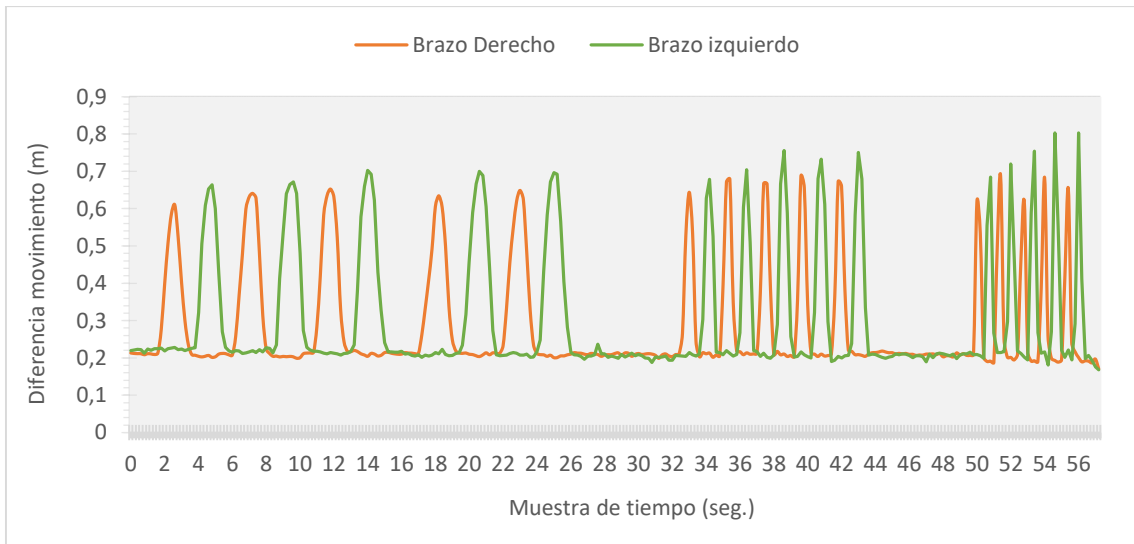


Figura 4.3: Gráfico del registro de datos con el equipo 2 y frecuencia de muestreo de 0,2 segundos

Tras ver los resultados de las 2 mediciones comparativas, se intentó realizar una tercera medición con el equipo 1 bajando el tiempo de muestreo a 0,2 segundos. Debido a la poca potencia del ordenador al ejecutar de manera simultánea varias tareas, el equipo se saturó acumulando procesos y colapsando el ordenador. Por lo tanto, no es posible establecer un valor de 0,2 segundos en el periodo de muestreo del programa sin que se vea afectado el rendimiento de los programas que se ejecutan simultáneamente en el mismo.

Tras los resultados obtenidos se pueden establecer las siguientes conclusiones:

- La potencia del procesador no es determinante de forma directa a la hora de registrar datos de uso en el fichero *.csv* pero sí que lo es a la hora de poder ajustar el valor de periodo de muestro y de gestionar correctamente varias tareas de forma simultánea.
- El periodo entre muestras establecido en 0,5 no es suficiente para capturar de forma correcta en el registro movimientos realizados de forma rápida y tiene dificultades para registrar la máxima amplitud de movimientos realizados a una velocidad normal. Sin embargo, es un valor suficiente para registrar movimientos a velocidad lenta.
- Establecer un periodo de muestro de 0,2 segundos es suficiente para registrar todos los movimientos realizados a distintas velocidades y su máxima amplitud, pero afecta notablemente al rendimiento del programa en equipos poco potentes.

Buscando la máxima compatibilidad de Mokey en todo tipo de equipo y que este no sea afectado en su rendimiento, se decide mantener el valor del periodo de muestro en 0,5 segundos, tiempo suficiente para captar la amplitud completa de los movimientos realizados a una velocidad normal en equipos poco potentes, tal y como se demuestra en la prueba. De esta forma se permite el uso del programa en ordenadores poco potentes, sin afectar al rendimiento normal del mismo. Se abre la posibilidad de modificar este valor en un futuro a medio y largo plazo.

4.5. Limitaciones del programa

En cuanto a las limitaciones que se han observado en la Interfaz Natural Mokey durante el desarrollo de las pruebas realizadas, se pueden definir las siguientes:

- Se produce limitación en cuanto a las aplicaciones o programas compatibles debido a que, con Mokey, no existen suficientes movimientos configurables como para simular todo el teclado, imposibilitando así la escritura de texto. Además, no es compatible con programas que requieran forzosamente movimiento del ratón ni programas que consuman gran parte de los recursos del ordenador empleado. Existe limitación también con programas que requieran pulsos de tecla prolongada debido a la función empleada para simular los eventos de teclado detallada en [1].
- En algunos casos, los movimientos propuestos en el programa no son suficientes, en especial para usuarios con limitación en los movimientos de brazos hacia fuera.
- Se observan problemas en la detección de los brazos en usuarios en silla de ruedas motorizadas. El sistema no reconoce de forma correcta el brazo relativo al lado donde se encuentra el mando de la silla de rueda, confundiéndolo con el mismo.
- Los movimientos relativos al tronco no son adecuados para personas con poca estabilidad. En ciertos casos, esto se soluciona con valores altos del parámetro *displacement* para evitar ejecutar movimientos indeseados.
- Es necesario ejecutar el programa en un entorno libre de objetos y personas ajenas para evitar frustración en los usuarios debido a interferencias y falsas detecciones durante la ejecución del programa.
- Se observan problemas en los movimientos grabados con los brazos frente al cuerpo debido a limitaciones de Kinect y la detección de puntos.
- Problema en la cola de procesos en ordenadores no muy potentes. Más detalle en [1].

Capítulo 5. CONCLUSIONES Y LÍNEAS FUTURAS

Tras el desarrollo de la Interfaz Natural Mokey y la realización de las pruebas del capítulo 4, en este capítulo se detallan las conclusiones, limitaciones y líneas futuras de desarrollo que han resultado del análisis de dichas pruebas. En [1] se realizaron las pruebas y conclusiones referentes a las pruebas de rendimiento realizadas al programa que completan las conclusiones de este documento.

5.1. Conclusiones

Es posible obtener conclusiones relacionadas a dos aspectos diferentes. En primer lugar, conclusiones relacionadas con las características finales del programa de la Interfaz gráfica y, en segundo lugar, conclusiones tras las pruebas *in situ* realizadas, tanto desde el punto de vista de experiencia de configuración, como desde el punto de vista del resultado real y la compatibilidad del programa. Las características técnicas logradas finalmente en Mokey son:

- Se permite el uso, mediante movimientos, de aplicaciones y programas en Windows que puedan ser controladas por medio del teclado del ordenador. Sin embargo, no es posible escribir texto debido a no tener movimientos suficientes predefinidos.
- Se permite al usuario la elección de 12 movimientos predefinidos configurables para simular los eventos establecidos de teclado.
- Se permite al usuario la configuración de 4 movimientos grabados adicionales a los predefinidos con el objetivo de ampliar el rango de movimientos seleccionables. Para usuarios en silla de ruedas, debido a las interferencias de esta, no funciona correctamente esta función. Falta un modo dedicado a la grabación de movimientos en silla de ruedas.
- Se ha conseguido eliminar el proceso de calibración detallado en [1]. Todos los movimientos se establecen mediante las coordenadas relativas de los puntos reconocidos por Kinect.
- Se permite al usuario la personalización del retardo o periodo entre eventos del teclado para cada movimiento configurado.
- Se permite al usuario la personalización de la amplitud de movimiento necesaria para reconocer cada uno de los movimientos configurados.
- Se permite la configuración de un modo destinado, especialmente, a usuarios en sillas de ruedas. Este modo cambia el punto de referencia para detectar los movimientos, de forma que al activarlo se posibilite el uso de todos los movimientos configurables en igualdad de condiciones frente a cualquier otro usuario.
- Se permite el guardado de sesiones personales y datos de usuario con el objetivo de almacenar toda la información de cada usuario en un mismo espacio de trabajo. Se permite al usuario guardar y cargar archivos de configuración de la interfaz gráfica con el objetivo de recuperar los ajustes establecidos previamente. Una vez que el usuario encuentra la configuración deseada, la puede almacenar y reutilizar.
- Se permite el registro de actividad del usuario de forma que se exporten los datos de los movimientos del usuario tras el uso del programa. La exportación se realiza en un fichero *.csv* y ayuda a establecer un control y un seguimiento por parte de asistentes o terapeutas. Hay que tener en cuenta que, para movimientos muy rápidos, el programa podría no registrar la amplitud máxima de cada movimiento.
- Cada movimiento se ejecuta de forma autónoma en un hilo independiente, de forma que posibilita la edición de los parámetros de los movimientos en tiempo real.

Referente a los aspectos relacionados con la interfaz gráfica y las pruebas realizadas en el capítulo 4, se establecen las siguientes conclusiones:

- Las pruebas determinan que la interfaz gráfica diseñada en Mokey es simple y de fácil configuración. Al contrario que con FAAST, usuarios sin experiencia previa con el programa pueden establecer su propia configuración en menos de 90 segundos.
- Según las pruebas con voluntarios, en la mayoría de ocasiones, los movimientos predefinidos son suficientes para establecer las configuraciones requeridas para manejar un programa. No ocurre así en personas en silla de ruedas con movilidad limitada en el tronco y brazos.
- El rango de ajuste de los parámetros de *delay* y *displacement* son suficientes en la mayoría de casos.
- Ante los usuarios que, por su discapacidad, no pueden repetir movimientos exactos, el programa ha respondido correctamente debido al diseño de los algoritmos de los movimientos detallados en [1].
- Para el correcto uso del programa, se hace necesario un periodo de adaptación por parte de los usuarios, en especial en usuarios con discapacidad.
- El registro de datos de movimiento durante el uso de una aplicación podría permitir a un asistente o terapeuta analizar la efectividad de los movimientos realizados, p.ej. como ejercicio.

Los resultados de rendimiento se publicaron en el *International Symposium on Consumer Electronics (ISCE)* en 2015 [2] y las pruebas con usuarios junto con una descripción completa del sistema final se publicará en junio de 2017 en la revista internacional *Assistive Technology* [3].

5.2. Líneas futuras

Tras las conclusiones extraídas durante el análisis del programa y las pruebas realizadas, se pueden observar varios puntos a tratar. En [1] se proponen líneas futuras relacionadas con el rendimiento y optimización de las funciones que afectan a los procesos. A continuación, se muestran las líneas de mejora en relación a lo tratado en este documento:

- Limitar las interferencias referentes al espacio de trabajo que capta Kinect mediante su control con funciones en el código fuente del programa. Esto evitaría falsas detecciones y frustración de los usuarios al utilizar el programa.
- Estudio de un procedimiento para que el programa detecte dos usuarios de forma simultánea. Especialmente útil en aplicaciones que pueden ser utilizadas por dos usuarios a la vez como juegos de dos personas o, en rehabilitación, en aplicaciones diseñadas en las que pueda interferir un asistente o terapeuta en la realización de ejercicios.
- Ampliar la lista de movimientos predefinidos en el programa para su elección por el usuario. Interesante añadir más movimientos con el tronco del cuerpo o con los brazos de forma que los usuarios en silla de ruedas con poca movilidad tengan más posibilidades de configuración, independientemente de la grabación de movimientos.
- Ampliar el número de movimientos posibles a grabar y crear un modo de grabación de movimientos destinado al usuario en silla de ruedas en el cuál se modifiquen los huesos tomados como referencia en el proceso de grabación.

- Añadir la posibilidad de capturar movimientos grabados con toda la trayectoria de una extremidad concreta, de forma que se detecten los mismos movimientos y su trayectoria dentro de una determinada tolerancia. Además, visualizar el proceso de grabación.
- Permitir la simulación de eventos del ratón asociados a los movimientos. De esta forma se conseguiría un programa más versátil y con mayores posibilidades de adaptación a otras aplicaciones abriendo la posibilidad de controlar un teclado en pantalla para escribir texto. Además, el cambio de la función empleada en el envío de eventos de teclado -*Sendkeys [1]*- introduciría mayores posibilidades en el envío de eventos de pulso continuo a terceras aplicaciones.
- Mejorar la interfaz gráfica de usuario de forma que el usuario se pueda ver reflejado en un esqueleto e imagen real en una ventana del programa. Sería de gran ayuda a la hora de configurar los parámetros de cada movimiento.
- Introducir la posibilidad de configurar los parámetros insertando un valor por teclado, además de los deslizadores.
- Añadir un botón “Stop Mokey” para poder parar la interfaz sin cerrarla y grabar los ficheros de una sesión para después p.ej. seguir usando programa.
- Ampliar las posibilidades de uso con software de terceros más sofisticado.
- Existe un gran potencial de desarrollo de aplicaciones y programaciones para usar Mokey en ámbitos de rehabilitación y control del ejercicio para personas con discapacidad o con problemas de movilidad. El desarrollo de aplicaciones que puedan ser utilizadas como apoyo a asistentes o terapeutas de la misma forma que lo hace el registro de datos de uso que está diseñado en Mokey. La compatibilidad con Adobe Flash permite la creación de pequeñas aplicaciones destinadas a este fin.
- Introducir el control por voz ampliaría las posibilidades de configuración del programa pudiendo configurar órdenes o eventos de teclado mediante comandos de voz.
- Adaptar el programa a la segunda versión de Kinect, esto permitiría la detección de gestos faciales o gestos con las manos de forma más precisa y ampliaría de forma considerable las posibilidades de configuración de Mokey. Además, la eficacia y precisión de Mokey se vería notablemente favorecida.

REFERENCIAS

- [1] M. López, *Desarrollo del teclado virtual 'MoKey' basado en gestos para personas con movilidad reducida: capa de sistema*, Proyecto Fin de Grado, Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, Universidad Politécnica de Madrid, septiembre 2015.
- [2] M. Eckert, M. López, C. Lázaro, J. Meneses, J.F. Martínez Ortega, "MoKey – A Motion based Keyboard Interpreter", *IEEE International Symposium on Consumer Electronics (ISCE)*, 2015.
- [3] M. Eckert, M. López, C. Lázaro, J. Meneses, J.F. Martínez Ortega, "MoKey: A Versatile Exergame Creator for Everyday Usage", *Assitive Technology*, accepted, to be published in June 2017.
- [4] World Health Organization, *World Report on Disability*, [disponible en línea: http://whqlibdoc.who.int/publications/2011/9789240685215_eng.pdf?ua=1.] acceso Agosto 2016.
- [5] Microsoft Corporation, *Kinect Manual*, Microsoft Corporation, 2010.
- [6] Microsoft Corporation, "Microsoft Accessibility", [disponible en línea: <https://www.microsoft.com/en-us/Accessibility>], acceso febrero 2017.
- [7] Evolve Company, "Kinect for Windows Software", [disponible en línea: <http://www.evolve.com/kinect-for-windows-software.htm>], acceso Febrero 2017.
- [8] E. a. Suma, B. Lange, A. S. Rizzo, D. M. Krum, y M. Bolas, "FAAST: The Flexible Action and Articulated Skeleton Toolkit," *IEEE Virtual Reality Conference*, pp. 247–248, marzo 2011.
- [9] USC Institute for Creative Technologies, "Flexible Action and Articulated Skeleton Toolkit (FAAST)", [disponible en línea: <http://projects.ict.usc.edu/mxr/faast/>], acceso Febrero 2017.
- [10] Microsoft Corporation, "Kinect Hardware", [disponible en línea: <https://developer.microsoft.com/en-us/windows/kinect/hardware>], acceso Febrero 2017.
- [11] M. Elgendi, F. Picon, N. Magnenat y D. Abbott, "Arm movement speed assessment via a Kinect camera: A preliminary study in healthy subjects", *BioMedical Engineering OnLine*, vol. 13, Julio 2014.
- [12] H. Jiang, B. S. Duerstock y J. P. Wachs, "Integrated gesture recognition-based interface for people with upper extremity mobility impairments", in *Advances in Human Aspects of Healthcare*, editor Duffy VG: CRC Press, pp. 546–555, 2012.
- [13] S. Koenig, A. Ardanza, y C. Cortes, "Introduction to Low-Cost Motion-Tracking for Virtual Rehabilitation", in J. L. Pons & D. Torricelli (Eds.), Springer Berlin Heidelberg, *Emerging Therapies in Neurorehabilitation*, vol. 4, pp. 287–303.
- [14] S. M. Pool, J. M. Hoyle, L. A. Malone, L. Cooper, C. S. Bickel, G. McGwin, J. H. Rimmer, & A. W. Eberhardt, "Navigation of a virtual exercise environment with

- Microsoft Kinect by people post-stroke or with cerebral palsy”, *Assistive Technology*, 28(4), pp. 225-232, 2016.
- [15] Open Natural Interaction, “OPENNI SDK”, [Disponible en línea: <http://openni.ru/openni-sdk/>], acceso noviembre 2016.
- [16] Microsoft Corporation, “Kinect for Windows SDK”, [Disponible en línea: <https://msdn.microsoft.com/en-us/library/hh855347.aspx>], acceso febrero 2017.
- [17] Microsoft Corporation, “Kinect for Windows Sensor Components and Specifications”, [Disponible en línea: <https://developer.microsoft.com/en-us/windows/kinect/hardware>], acceso febrero 2017.
- [18] S. Brunner, “Using Microsoft Kinect Sensor to perform commands on virtual objects”, Final Project Thesis, *Master in computer science*, 2012.
- [19] Microsoft Corporation, “Features Kinect 2.0”, [Disponible en línea: <https://msdn.microsoft.com/library/dn782025.aspx>], acceso febrero 2017.
- [20] F. Lamberti, A. Sanna, G. Paravati, C. Demartini, y P. Torino, “Endowing Existing Desktop Applications with Customizable Body Gesture-based Interfaces,” *International Conference on Consumer Electronics*, pp. 558–559, 2013.
- [21] S. Lee y S. Oh, “A Kinect Sensor based Windows Control Interface”, *Int. J. Control Autom.*, vol. 7, no. 3, pp. 113–124, 2014.
- [22] A. Kar, “Skeletal Tracking using Microsoft Kinect”, 2011.
- [23] L. Xia, C. Chen y J.K. Aggarwal, “Human Detection Using Depth Information by Kinect”, *IEEE Computer Society Conference*, pp. 15-22, junio 2011.
- [24] C. Sinthanayothin, N. Wongwaen and W. Bholsithi, “Skeleton tracking using Kinect Sensor & Displaying in 3D Virtual Scene”, *IJACT*, Vol. 4, No. 11, pp. 213-223, 2012.
- [25] M. Martínez, F.J. Díaz, A. Tejero, F. Perozo, M. Antón y D. González, “Monitorización del cuerpo humano en 3D mediante tecnología Kinect”, *SAAEI XVIII Edition*, pp. 747-752.
- [26] M. Elgendi, F. Picon y N. Magnenat, “Real-Time Speed Detection of Hand Gesture using Kinect”, *Annual Conference on Computer Animation and Social Agents*, mayo 2012.
- [27] Microsoft Corporation, “C#”, [Disponible en línea: <https://msdn.microsoft.com/es-es/library/kx37x362.aspx>], acceso febrero 2017.
- [28] Organización Internacional de Normalización, “ISO/IEC 23270:2006”, septiembre 2006.
- [29] Microsoft Corporation, “Visual Studio”, [Disponible en línea: <https://www.visualstudio.com/es/>], acceso marzo 2017.

- [30] Microsoft Corporation, “System.Windows Namespace”, [Disponible en línea: <https://msdn.microsoft.com/en-us/library/system.windows.aspx>], acceso marzo 2017.

Anexo 1. PRESUPUESTO

En este anexo se elabora un presupuesto de los costes generales del proyecto, tanto a nivel de material como a nivel de horas de trabajo dedicadas al desarrollo del trabajo y a la redacción de la memoria.

Materiales:

- Equipo 1.....480 €
 - ✓ Intel® Core 2 6400 a 2.13 GHz
 - ✓ 4 GB RAM DDR3
 - ✓ 500 GB Disco Duro 5400 rpm
 - ✓ Intel Graphics

- Equipo 2.....1220€
 - ✓ Intel® Core i7-4790 a 3.6 GHz
 - ✓ 16 GB RAM DDR3
 - ✓ 1000 GB Disco Duro 5400 rpm
 - ✓ 128 GB SSD
 - ✓ 4 GB ATI Radeon

- Microsoft Kinect 360 1.0.....140 €

- Adaptador Microsoft Kinect 360 PC compatible20 €

Horas de trabajo:

- 6 meses de trabajo prácticas curriculares
 - ✓ 6 créditos ECTS (45 horas/crédito)270 horas
totales

- 8 meses de trabajo Prácticas de Colaboración MECD
 - ✓ 15 horas semanales480 horas
totales

- 6 meses realización pruebas y redacción informe
 - ✓ 6 horas semanales144 horas
totales

Anexo 2. MANUAL DE USUARIO DE MOKEY

A lo largo del presente anexo se elabora un manual de instalación rápida y de uso de la aplicación Mokey con el fin de orientar y guiar a los usuarios en su primer uso del programa. Este manual se presenta como actualización al presentado por Marcos López García en el Anexo 1 de [1] debido a los cambios introducidos en la versión presentada en la actual memoria.

El manual está orientado en base a la instalación en ordenadores con el sistema operativo **Windows 10** o inferior del **kit de desarrollo SDK 1.8** de Kinect para el control de la cámara **Microsoft Kinect 360 1.0**. Puesto que es un programa orientado al entorno Windows, tan solo es posible instalarlo en dicho sistema siendo compatible en Windows 10 o inferiores sin asegurar su compatibilidad en futuras versiones.

A2.1. Instalación de Mokey

Antes de ejecutar el archivo *Mokey.exe* relativo al programa, es necesario tener correctamente instalados en el sistema los *drivers* y el kit de desarrollo SDK 1.8. Se recomienda, al instalar el kit de desarrollo, tener desconectada la cámara Microsoft Kinect. El kit de desarrollo, junto con los drivers, se puede descargar la página oficial de Microsoft [2].

Una vez instalado el SDK 1.8 y los *drivers* de Kinect, se puede comprobar que el ordenador detecta correctamente la cámara al conectarla. De ser así, el piloto de luz verde del dispositivo quedará fijo, sin parpadear.

Tras comprobar que el equipo detecta correctamente el dispositivo Kinect, para abrir Mokey tan solo hay que ejecutar el archivo *Mokey.exe* sin necesidad de ninguna instalación adicional.

A2.2. Configuración de Mokey

A continuación, se detallan los pasos previos a la utilización de Mokey:

- 1) **Conectar la cámara Microsoft Kinect a un puerto USB libre del ordenador** y comprobar que esta ha sido correctamente detectada. Comprobar que en la barra de tareas de Windows no aparece ningún driver instalado y que es posible extraer de forma segura el dispositivo Kinect. De no ser así, es posible que no se hayan instalado correctamente los drivers y haya que repetir el proceso de instalación.
- 2) **Ejecutar el archivo “Mokey.exe”** haciendo doble clic sobre el mismo para ir a la primera ventana de Mokey, la ventana de bienvenida e inicio de sesión mostrada en la figura A2.1.

En esta primera pantalla de bienvenida se debe introducir, en el cuadro de texto habilitado, el nombre de aquella persona que va a utilizar Mokey. Este nombre será el relativo a la sesión en la cual Mokey va a almacenar todos los datos personales, archivos de configuración y registros de uso personalizados para ese mismo usuario.

Si el usuario es nuevo, creará una carpeta destinada al mismo usuario en la carpeta “Mokey” ubicada en la raíz del disco duro del sistema operativo. Si el nombre de sesión ya existe, Mokey cargará toda la información previamente almacenada en dicha sesión.

Una vez introducido el nombre de la sesión, pulsar el botón “Start” para ir a la siguiente ventana de configuración.



Figura A2.1: Pantalla de bienvenida e inicio de sesión de Mokey

- 3) **Selección del archivo de configuración:** La siguiente ventana hace referencia a la selección del archivo de configuración previamente guardado en la sesión iniciada. Si la sesión no tiene *presets* previamente guardados, se mostrará una ventana como la de la figura A2.2, sin ningún archivo para seleccionar. Al seleccionar un fichero de la lista, se muestran los comentarios asociados en la parte inferior de la ventana.

En caso de querer iniciar Mokey con la configuración por defecto, seleccionar en la lista la línea “New Sesion”.

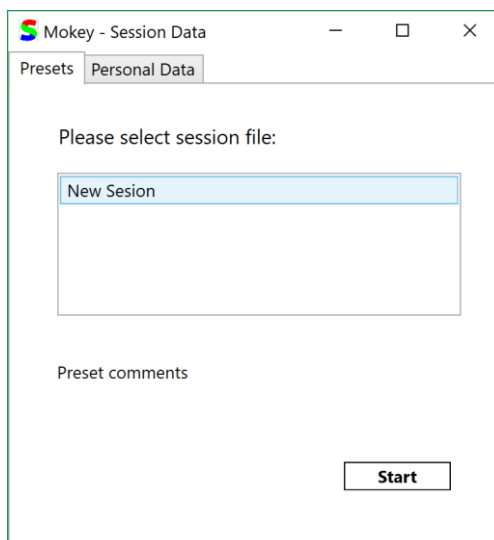


Figura A2.2: Pantalla de selección de archivo de configuración de Mokey

Una vez seleccionado el archivo de configuración, si se desea se puede ir a la segunda pestaña de la ventana denominada “Personal Data” explicado en el siguiente paso 4 o bien pulsar el botón “Start” e ir directamente al paso 5 de configuración de la ventana principal.

- 4) **Guardado de datos personales asociados a la sesión iniciada:** En esta segunda pestaña se pueden introducir, en los campos mostrados en la figura A2.3, los datos de nombre, apellidos, edad, discapacidad u observaciones. Una vez cumplimentada esta información, para guardarla hay que pulsar el botón “Save”.

Una vez guardados los datos personales, para continuar al siguiente paso hay que volver a la pestaña “Presets” y pulsar el botón “Start”.

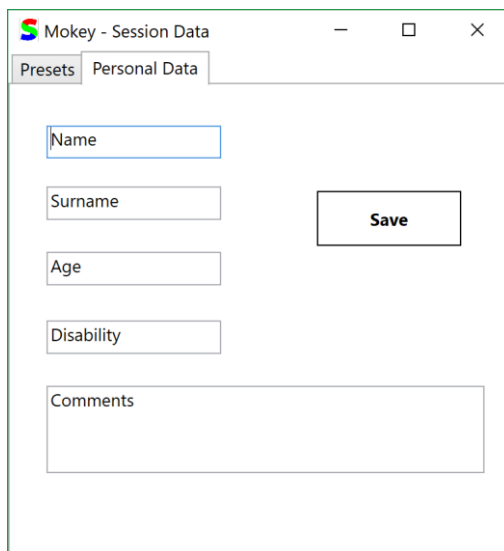


Figura A2.3: Pestaña de introducción de datos personales de la sesión de Mokey

- 5) **Pantalla principal de configuración de Mokey:** En caso de que la cámara Kinect no estuviera correctamente conectada, el programa lanza la ventana de advertencia mostrada en la figura A2.4. En caso contrario, una vez seleccionado el archivo de configuración y pulsado el botón “Start” en el paso anterior, el programa lanza la ventana principal de configuración de Mokey mostrada en la figura A2.5.

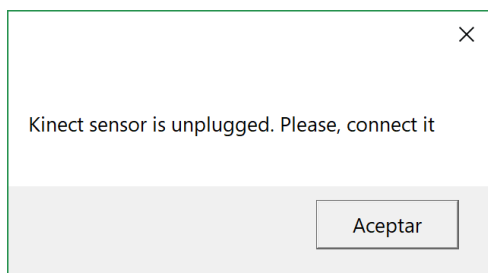


Figura A2.4: Mensaje de advertencia del programa al no detectar Kinect correctamente.

En la ventana principal mostrada en la figura A2.5 se encuentra el lugar donde se configuran todos los movimientos, ajuste de parámetros y teclas asociadas a cada uno de ellos que se desean configurar para el uso de Mokey. En los siguientes pasos se dan las instrucciones para configurar cada uno de sus módulos.

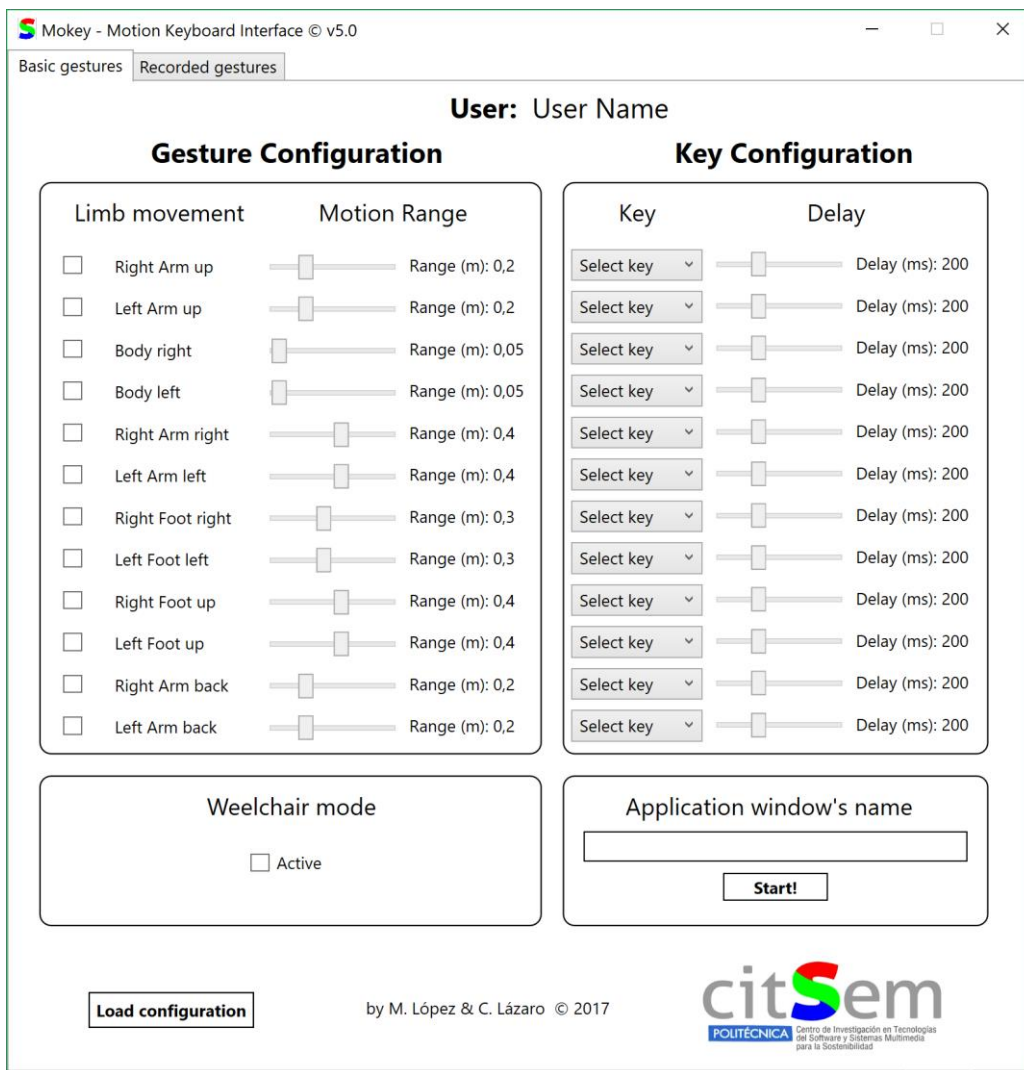


Figura A2.5: Ventana de advertencia del programa al no detectar Kinect correctamente.

- 6) **Seleccionar si se va a utilizar el Modo *Wheelchair*, especial para silla de ruedas:** Este modo optimiza el uso del programa para personas que permanezcan en silla de ruedas durante su utilización modificando, en los algoritmos de los movimientos, los huesos de referencia sobre los que se calculan los movimientos (Más detalle en el capítulo 3 de la memoria). Para su activación, seleccionar el cuadro de selección de la figura A2.6.



Figura A2.6: Selección modo *Wheelchair* en la ventana principal de configuración

- 7) **Determinar qué movimientos se van a utilizar:** En la figura A2.7 se muestra el módulo con la selección de los 12 movimientos predefinidos disponibles para su utilización. Más información sobre cada uno ellos en el capítulo 3 de esta memoria. Se debe activar el cuadro de selección situado a la izquierda del nombre del movimiento para hacer uso del mismo.

Gesture Configuration

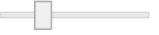
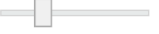

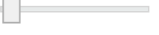







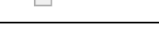
Limb movement	Motion Range
<input type="checkbox"/> Right Arm up	 Range (m): 0,2
<input type="checkbox"/> Left Arm up	 Range (m): 0,2
<input type="checkbox"/> Body right	 Range (m): 0,05
<input type="checkbox"/> Body left	 Range (m): 0,05
<input type="checkbox"/> Right Arm right	 Range (m): 0,4
<input type="checkbox"/> Left Arm left	 Range (m): 0,4
<input type="checkbox"/> Right Foot right	 Range (m): 0,3
<input type="checkbox"/> Left Foot left	 Range (m): 0,3
<input type="checkbox"/> Right Foot up	 Range (m): 0,4
<input type="checkbox"/> Left Foot up	 Range (m): 0,4
<input type="checkbox"/> Right Arm back	 Range (m): 0,2
<input type="checkbox"/> Left Arm back	 Range (m): 0,2

Figura A2.7: Selección de movimientos y ajuste del parámetro displacement.

- 8) **Ajustar el parámetro displacement de cada movimiento seleccionado:** En la figura A2.7, además de seleccionar los movimientos que se van a usar, también se establece el parámetro displacement para cada uno de ellos. Este valor se modifica deslizando el control deslizante en la posición del valor deseado, teniendo como guía para ello el valor numérico en la parte derecha de cada slider.

Es posible ajustar el valor entre 0 y 0.7 metros. Cuanto más grande sea el valor, más amplio debe ser el movimiento para ejecutar el evento de teclado correspondiente. Por defecto, este parámetro viene configurado con el valor óptimo para cada movimiento.

- 9) **Selección de tecla asociada a cada movimiento:** Tras seleccionar los movimientos, en el siguiente módulo mostrado en la figura A2.8 se debe seleccionar la tecla asociada a cada uno. En la figura A2.9 se muestra el desplegable alineado con cada movimiento para hacer selección de la tecla que se quiere simular de todas las opciones ofrecidas en el desplegable.

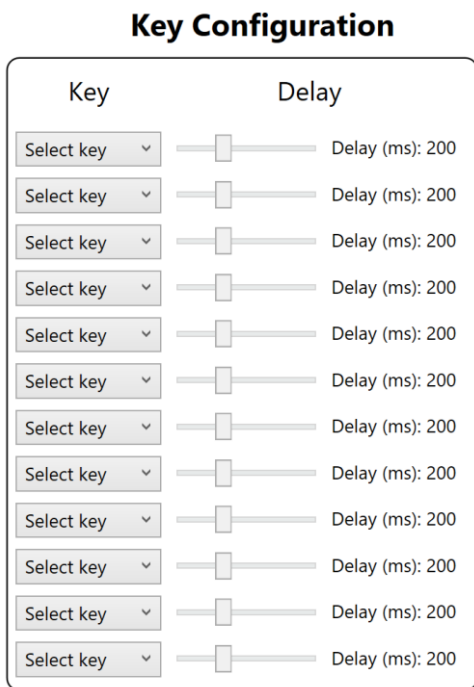


Figura A2.8: Selección de tecla asociada a cada movimiento y ajuste del parámetro delay.

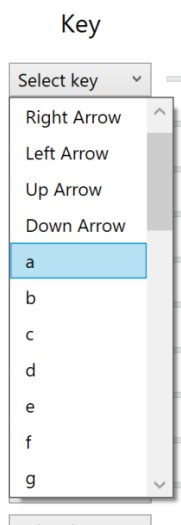


Figura A2.9: Desplegable de la selección de tecla asociada a cada movimiento.

- 10) **Ajustar el parámetro *delay* de cada movimiento seleccionado:** Para ello, en el control deslizante de la parte derecha de la figura A2.8, se pueden seleccionar valores del parámetro *delay* desde 0 hasta 600 milisegundos. Cuanto menor sea este valor, menor será el espacio de tiempo entre dos eventos de teclado consecutivos asociados al movimiento.

Como guía, a la derecha del control deslizante, se muestra el valor del *slider* en ese instante. En todos los movimientos, el valor por defecto es de 200 milisegundos.

- 11) **Indicar el nombre de la ventana que se va utilizar (opcional):** En el campo mostrado en la figura A2.10, opcionalmente, se puede escribir el nombre de la ventana sobre la que va a actuar Mokey. Al situarse el usuario frente a la cámara Kinect, el programa pondrá de forma automática en primer plano la aplicación cuyo título de ventana coincida con lo escrito en este campo.

El nombre de ventana que se debe escribir es el mismo que se muestra en la ventana de título de la aplicación, situada en la parte superior. Se muestra un ejemplo de un archivo de PowerPoint en la figura A2.11.

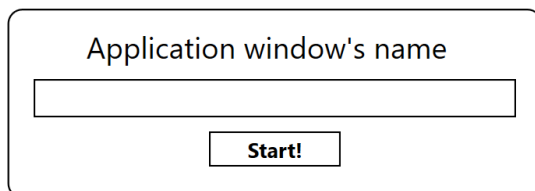


Figura A2.10: Módulo relativo a la aplicación controlada con Mokey.

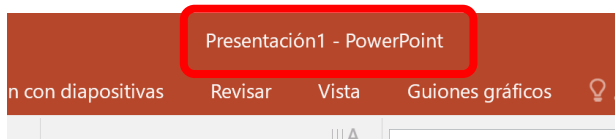


Figura A2.11: Ejemplo de título de ventana de un archivo de PowerPoint 2016.

- 12) **Seleccionar un archivo de configuración previamente guardado (opcional):** Para facilitar el uso de Mokey, de forma opcional, es posible cargar una configuración de la interfaz gráfica previamente guardada en sesiones previas. Para ello se debe pulsar el botón, mostrado en la figura A2.12, situado en la parte inferior izquierda de la pantalla principal de configuración de Mokey.

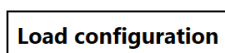


Figura A2.12: Botón relativo a la carga de un archivo de configuración de la interfaz gráfica.

Tras pulsar el botón se abre un cuadro de navegación de archivos donde se permite buscar y seleccionar el fichero *.txt* asociado a la configuración que se quiere cargar en el programa. Una vez cargado, la interfaz gráfica se actualizará acorde a la información del fichero seleccionado.

- 13) **Pulsar el botón “Start!” para comenzar la ejecución del programa:** Una vez se configuren todos los movimientos que se quiere utilizar y todos sus parámetros asociados, al pulsar el botón “Start!” mostrado en la figura A2.10 se ejecuta el programa.

Es importante señalar que antes de pulsar el botón, es conveniente tener ejecutándose en segundo plano la aplicación sobre la que actuará Mokey. De ser así, una vez pulsado el botón, la aplicación indicada en el paso numero 11 pasará a primer plano cuando Mokey detecte el usuario frente a Kinect.

- 14) **Para finalizar el funcionamiento de Mokey**, basta con cerrar la ventana principal de Mokey. De forma automática, el programa cerrará todos los hilos abiertos en segundo plano.
- 15) **Guardar archivo de configuración para futuros usos:** Una vez finalizado el uso de Mokey, cerrando la ventana principal de configuración, aparece la ventana mostrada en la figura A2.13. En ella, es posible guardar la configuración de todos los movimientos y parámetros asociados que se usaron por última vez en el programa pulsando el botón “Yes”. De esta forma, es posible guardar la configuración para volver a utilizarla en posteriores usos del programa.

En esta ventana es posible asignar nombre y comentarios al archivo de configuración se guardará. El programa guardará en el directorio de la sesión (C:\Mokey\Sesión\configs) el archivo de configuración en formato *.txt* según el nombre de archivo indicado. Si no se quiere guardar la configuración, tan solo hay que pulsar en el botón “No”.

No se recomienda la edición manual de los archivos de configuración guardados debido a que podrían quedar inservibles para futuros usos.

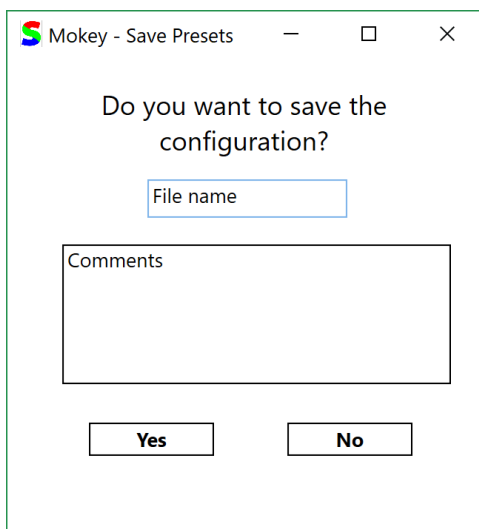


Figura A2.13: Ventana de guardado de archivos de configuración.

A2.3. Grabar movimientos personalizados

De forma opcional, es posible grabar tus propios movimientos si no se ha encontrado el deseado entre los 12 movimientos predefinidos en el programa. Se permite grabar hasta 4 movimientos diferentes. Para ello, en la segunda pestaña de la ventana principal de configuración mostrada en la figura A2.5 y denominada “Recorded gestures” se accede al módulo de grabación mostrado en la figura A2.14. Esta grabación se debe realizar antes del paso número 13.

Para el proceso de grabación es necesario que una persona ajena a la grabación supervise y ayude al usuario durante el procedimiento.

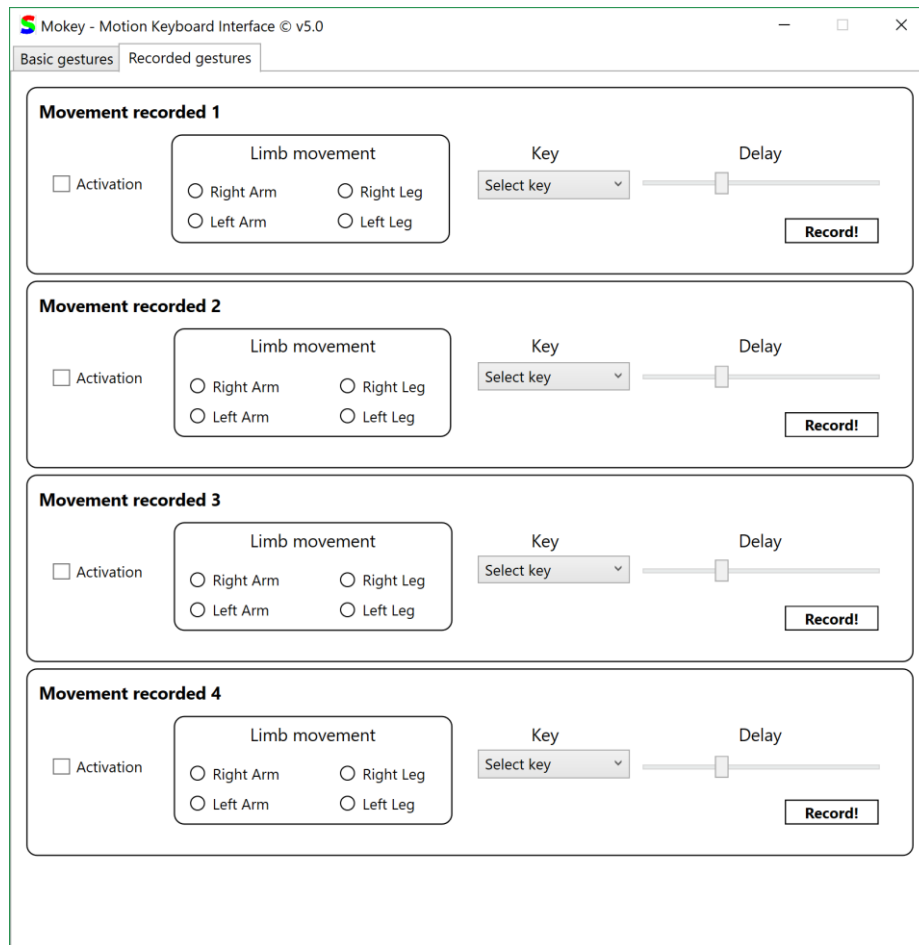


Figura A2.14: Ventana de configuración de la grabación de movimientos.

Los pasos relativos a la grabación de los movimientos son los siguientes:

1. **Seleccionar el módulo de grabación que se desea activar** en los cuadros de selección “Activation” de la parte izquierda. Cada movimiento grabado se asocia a un módulo de grabación propio.
2. **Seleccionar el miembro del cuerpo sobre el que se va a realizar la grabación.** Es posible elegir entre cuatro posibilidades: los dos brazos y las dos piernas tal y como se muestra en el cuadro titulado “Limb movement” en cada módulo.
3. **La persona que desea grabar el movimiento se debe colocar frente a Kinect.** Para realizar la grabación, tan solo es necesario realizar la posición final del movimiento a grabar. Una segunda persona ayudante debe pulsar el botón “Record!” del módulo de grabación correspondiente cuando el usuario que va a ser grabado tenga el miembro seleccionado en la posición final deseada. Una vez pulsado el botón, la grabación se realiza de forma instantánea mostrando un mensaje de información como el de la figura A2.15. Este mensaje muestra las coordenadas del punto grabado.

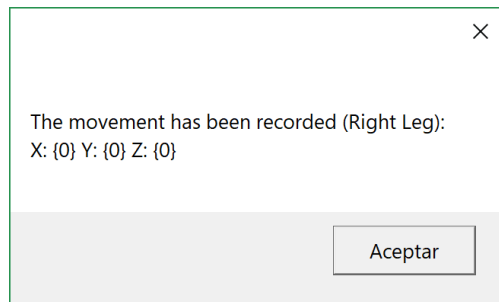


Figura A2.15: Mensaje informativo al realizar la grabación de un movimiento.

4. **Configurar la tecla asociada al movimiento.** De la misma que en los 12 movimientos predefinidos, hay una lista desplegable para realizar dicha selección.
5. **Configurar el parámetro *delay* asociado al movimiento.** Del mismo modo que con los 12 movimientos predefinidos, se debe establecer el tiempo en dos pulsaciones de teclado consecutivas. Para ello, cada módulo cuenta con un control deslizante.
6. **Continuar con el proceso de configuración desde el paso 11 del procedimiento del apartado A2.2.**

A2.4. Referencias

- [1] M. López, Desarrollo del teclado virtual ‘MoKey’ basado en gestos para personas con movilidad reducida: capa de sistema, Proyecto Fin de Grado, Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, Universidad Politécnica de Madrid, septiembre 2015.
- [2] Microsoft Corporation, “Kinect for Windows SDK”, [Disponible en línea: <https://msdn.microsoft.com/en-us/library/hh855347.aspx>] acceso febrero 2017.