

Definición de gestos 3D por usuario final: una experiencia práctica con Leap Motion

Jesús García González¹, Andrea Bellucci², Ignacio Aedo Cuevas³

¹ Estudiante de máster en Ciencia y Tecnología Informática, UC3M Leganés (Madrid), jgonzalez@alumnos.uc3m.es

² Postdoctorado, Departamento de Informática, UC3M Leganés (Madrid), abellucc@inf.uc3m.es

³ Catedrático, Departamento de Informática, UC3M, Leganés (Madrid), aedo@ia.uc3m.es

RESUMEN

Los sistemas de cómputo se han convertido en ubicuos, haciendo necesario el uso de nuevos paradigmas de interacción, como la interacción por gestos, y herramientas que permitan a los usuarios finales comunicarse con los ordenadores de una forma natural, ayudando a crear un flujo de interacción personalizado.

Sin embargo, la programación de nuevos gestos no es sencilla, ya que se necesita un conocimiento avanzado de las tecnologías hardware y lenguajes de programación. En este trabajo, se presenta la aplicación “Leap Learn” para la definición de gestos 3D a través de la técnica de “programación por ejemplos”, haciendo uso de la herramienta Leap Motion como dispositivo de entrada receptor de los gestos.

Palabras Clave — Desarrollo por usuario final, Interacción por Gestos, HCI, Leap Learn, Leap Motion.

I. INTRODUCCIÓN

Los dispositivos de cómputo ya no están solamente vinculados a entornos de trabajo, sino que se han convertido en ubicuos utilizándose también en la vida cotidiana, donde los clásicos dispositivos de entrada y salida, como el ratón o teclado, no cubren las necesidades de los usuarios en todos los contextos. A medida que los sistemas de cómputo van evolucionando, se hace necesaria una interacción más natural que permita la integración de la tecnología en el mundo real.

Esta evolución de los sistemas de interacción se manifiesta en avances como el reconocimiento de gestos, que en combinación con otras tecnologías de interfaz de usuario avanzadas, como los comandos de voz y reconocimiento de rostros, crean una experiencia de usuario que se esfuerza por comprender el “lenguaje” humano.

Existen actualmente diferentes soluciones comerciales, así como otros proyectos, que permiten el reconocimiento de gestos para transformarlos en operaciones de entrada del ordenador. Entre estas soluciones se encuentran la visión artificial mediante

cámaras [1], el reconocimiento mediante el dibujo de grafos con lápiz óptico o pluma indicadora [2], mediante ultrasonidos [3] o el análisis del reflejo de las señales Wifi [4]. Otras soluciones combinan dos o más tecnologías como los sistemas que utilizan el análisis de vídeo y señales infrarrojas [5] o acelerómetros y giroscopios [6].

A menudo, estos sistemas de reconocimiento de gestos no proporcionan a los usuarios la posibilidad de crear sus propias configuraciones y métodos de interacción. En este aspecto, los usuarios deben ceñirse a los elementos de interacción programados por defecto. No obstante, está demostrado que un vocabulario de gestos definidos por el usuario es más fácil de recordar que uno definido por el diseñador [7].

Sin embargo, la programación de nuevos gestos por parte de los usuarios finales no es sencilla. Se necesita el conocimiento previo de las tecnologías y lenguajes de programación de los cuales los usuarios carecen.

Para solventar este problema, se pone al alcance de los usuarios herramientas que les permitan la programación de sus propios gestos, aplicando el conjunto de técnicas y métodos de la Programación o Desarrollo por Usuario Final, *End User Development* (EUD).

El objetivo de esta investigación se centra en el desarrollo de la aplicación “Leap Learn”, que permite programar e interpretar nuevos gestos realizados por usuarios finales y ser asociados a diferentes acciones del sistema operativo, siguiendo la técnica de EUD: programación mediante ejemplos.

El dispositivo de entrada para la detección de los gestos es la herramienta “Leap Motion”, un sistema que utiliza la combinación de la visión artificial mediante cámaras y el análisis de señales infrarrojas para detectar los gestos realizados con brazos y manos. Una de las ventajas que presenta Leap Motion, frente a otras tecnologías, se encuentra en la eficacia de la interpretación de los gestos de la mano, con mayor resolución y definición que sus competidores [8].

1 Definición de gestos 3D por usuario final.
Una experiencia práctica con Leap Motion.

Para estudiar la usabilidad de la aplicación "Leap Learn", se ha utilizado una muestra de usuarios significativa que ha permitido realizar un análisis cualitativo del caso práctico mediante la escala *System Usability Scale* (SUS).

A lo largo de este trabajo se establecen las siguientes secciones: II Marco del proyecto: sitúa el trabajo dentro del contexto del desarrollo por parte de usuario final mediante gestos. III Trabajos relacionados: se identifican proyectos e ideas actuales relacionadas con el objetivo de este proyecto. IV Propuesta de solución: solución práctica desarrollada de acuerdo con la propuesta. V Evaluación de la aplicación: evaluación cualitativa de la usabilidad de la aplicación "Leap Learn". VI Conclusiones: conclusiones y futuras líneas de investigación del proyecto.

II. MARCO DEL PROYECTO

La EUD, también llamado *End User Programing* (EUP) es "un conjunto de métodos, técnicas y herramientas que permiten a los usuarios actuar como desarrolladores de software en algún momento para crear, modificar o extender una característica de software" [9]. El valor de la EUD reside en que los usuarios finales conocen su propio contexto y necesidades mejor que nadie, y muchas veces tienen conciencia en tiempo real de los cambios en sus respectivos dominios [10].

Sin embargo, la EUD es intrínsecamente diferente al desarrollo de software tradicional. Los usuarios finales no suelen tener formación en lenguajes de programación y procesos formales de desarrollo. Por otra parte, a menudo carecen de tiempo o motivación para aprender estas técnicas tradicionales, ya que los usuarios finales suelen realizar el desarrollo de tareas con el fin de lograr un objetivo a corto o medio plazo, en lugar de crear un software duradero. En consecuencia, el apoyo a la EUD requiere proporcionar las herramientas adecuadas y procesos de desarrollo que sean muy fáciles de utilizar, de rápido aprendizaje e integración en la práctica del dominio [11].

Dentro de las diferentes técnicas de la EUD se encuentra la Programación mediante Ejemplos, *Programming by Example* (PbE), que reduce la necesidad al usuario de aprender las abstracciones de un lenguaje de programación clásico, y desarrolle la programación del sistema mediante la introducción de algunos ejemplos de los resultados y/o operaciones que se desean conseguir. Nuevos datos pueden ser entonces introducidos de forma automática para que el sistema de PbE infiera las abstracciones correctas que producen esta salida.

La aplicación "Leap Learn" construida en este proyecto sigue la técnica de PbE para la generación de gestos y su posterior asociación a diferentes acciones del sistema.

Se entiende el reconocimiento de gestos como un tema de tecnología y ciencias de la computación que tiene como objetivo interpretar los gestos humanos a través

de algoritmos matemáticos. Los gestos pueden provenir de cualquier movimiento corporal o estado. Utilizando el concepto de reconocimiento de gestos, es posible señalar con el dedo a la pantalla del ordenador - por ejemplo - para que el cursor se mueva en consecuencia, complementando dispositivos de entrada convencionales, tales como ratones, teclados y pantallas táctiles.

A la hora de involucrar a los usuarios finales en la programación de tareas o actividades, los sistemas de interacción avanzada como la interpretación de gestos asumen un papel fundamental. La comunicación por gestos está intrínsecamente relacionada con la forma en la que los humanos se comunican entre sí, se utilizan para incrementar la efectividad de la comunicación, lo cual acerca a los usuarios a comunicarse con las máquinas de forma más natural [12].

III. TRABAJOS RELACIONADOS

La literatura incluye trabajos en curso en el campo de la EUD y la interacción por gestos que han sido organizados alrededor de tres perspectivas en este proyecto. Trabajos sobre el paradigma EUD y la técnica PbE relacionados con el reconocimiento de gestos, evolución del algoritmo de reconocimiento \$1 que ha sido adaptado para la aplicación Leap Learn, y en tercer lugar trabajos sobre herramientas de PbE en el ámbito de la interacción por gestos.

Las nuevas tecnologías que han irrumpido en el mercado relacionadas con el reconocimiento de gestos, ponen de manifiesto la, cada vez mayor, implantación en la sociedad de estos sistemas de interacción natural [13].

Los usuarios finales se benefician de paquetes de software extensibles diseñados específicamente para cada dominio [14]. En el caso de EUD, los usuarios repiten los procesos de programación que les han sido mostrado anteriormente para adaptar el software según el dominio de su conocimiento [15]. La idea de crear entornos de programación que sean más naturales ha hecho necesario el estudio sobre cómo la gente piensa acerca de las tareas de programación, a la hora de crear programas, cómo tratar de encontrar y corregir errores en los mismos [16]. En este trabajo se aplica la técnica PbE [17] [18] de EUD como paradigma que permite la programación de procesos complejos por parte de usuarios finales, permitiendo la evolución de los sistemas y servicios interactivos mediante la simplificación de los mismos. En la aplicación "Leap Learn" los nuevos gestos son interpretados por el algoritmo de inferencia siguiendo este patrón, donde los usuarios proveen de varios ejemplos del comportamiento de la aplicación al sistema.

Para poder interpretar las interacciones mediante gestos que los usuarios realizan en la aplicación "Leap Learn", es necesario utilizar algoritmos de inferencia que reconozcan estas interacciones. El algoritmo "\$1

recognizer” ha sido creado para reconocer gestos en dos dimensiones realizados sobre dispositivos con pantallas táctiles [19]. \$1 es un clasificador que hace uso del algoritmo del vecino más cercano con una función de puntuación euclidiana. \$1 tiene un porcentaje de precisión sobre el 97% con solamente 1 gesto grabado, y el 99% para 3 o más gestos grabados. Una segunda versión del algoritmo llamado “\$1 Protractor” [20] se ejecuta un 40% más rápido y consume menos memoria. Ambos algoritmos han sido adaptados en la aplicación “Leap Learn” para el reconocimiento de los gestos en 3D.

Herramientas basadas en la PbE como Gesture Coder [21], Proton[22], Touch [23] y Gesture Studio [24] han sido diseñadas para proporcionar las herramientas necesarias a los usuarios para crear sus propios gestos en dispositivos de entrada táctiles en dos dimensiones.

Mientras trabajos basados en modelos estadísticos tales como Modelos Ocultos de Markov [25], algoritmos de condensación [26], altamente sofisticados, o la propia herramienta Proton, permiten a los desarrolladores crear gestos multitouch especificando expresiones regulares en un editor gráfico.

En la aplicación desarrollada Leap Learn los usuarios no necesitan tener ningún conocimiento técnico para la creación de nuevos gestos en 3 dimensiones, tratando la algoritmia de inferencia de forma opaca para los usuarios. Esta aproximación es quizás conceptualmente similar a los métodos de reconocimiento de gestos basados en máquinas temporales de espacios finitos [27], aunque los estados se representan como una simple lista de texto que representan los gestos (espacio-temporales) introducidos.

En esta línea, en el salto a la interacción por gestos mediante interacción “sin toque” se encuentran Hotspotizer [28] y Skeleton toolkit [29]. Hotspotizer en particular, permite a los usuarios sin conocimientos de programación informática diseñar, visualizar y guardar, haciendo uso de Microsoft Kinect, los gestos realizados con el cuerpo completo. Estos gestos se asignan a comandos del teclado pudiendo utilizarse para controlar cualquier aplicación que se ejecuta en la plataforma Microsoft Windows. La aplicación Leap Learn está inspirada en Hotspotizer, diseñada para trabajar con Leap Motion y la plataforma Mac.

IV. PROPUESTA DE SOLUCIÓN

Para cubrir el objetivo de este proyecto, se ha creado una aplicación llamada “Leap Learn”, la cual permite al usuario final programar sus propios gestos en tres dimensiones y asociarlos a eventos del teclado del ordenador para realizar diferentes acciones.

La aplicación ha sido desarrollada en JavaScript, utilizando el API de Leap Motion para la recepción de los datos de entrada. Para inferir los gestos introducidos se ha modificado el algoritmo de reconocimiento “\$1 recognizer” para habilitar el reconocimiento de gestos en

tres dimensiones. Por otra parte, se ha desarrollado un módulo específico en node.js, llamado mac-keyboard [30], capaz de enviar señales del teclado al sistema operativo simulando la pulsación de las teclas. Así mismo, es posible descargar el código de Leap Learn desde la cuenta en *Github* del proyecto [31].

Leap Motion frente a otros dispositivos define un área de observación de los gestos más pequeña y de más alta resolución [32] lo que permite el seguimiento de los dedos de las manos de forma más precisa.

Funcionamiento de Leap Motion.

Leap Motion es un periférico USB diseñado para ser situado en un escritorio físico, mirando hacia arriba. El campo de visión es una superficie cónica, ver Figura 1. que se extiende desde aproximadamente 2.5cm a 60cm por encima del dispositivo..

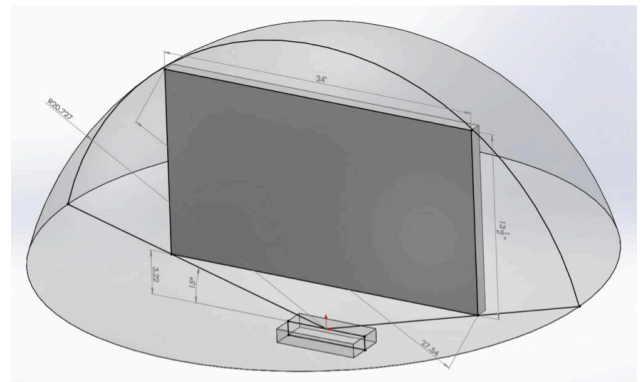


Figura 1, Área 3d de reconocimiento de Leap Motion.

El uso de dos cámaras de infrarrojos monocromáticas y tres LEDs infrarrojos, Figura 2, permite al software del controlador generar un patrón 3D a partir de los marcos 2D obtenidos por ambas cámaras.



(a)



(b)

Figura 2, desensamblado del Leap Motion (a). dos cámaras captan los patrones 3D. Tres diodos infrarrojos proporcionan la iluminación necesaria (b)

El API de Leap Motion pone a disposición de los desarrolladores 4 gestos básicos como se muestran en la Figura 3. Sin embargo, estos gestos predefinidos no cubren las necesidades del objetivo de este proyecto, dado que determinan la manera en la que los usuarios deben interactuar con el ordenador, y no proporciona herramientas para la creación de nuevos gestos.

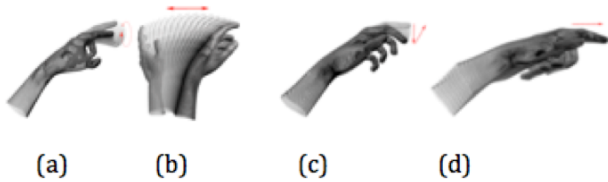


Figura 3, gestos definidos por el API de Leap Motion: Círculo (a), deslizamiento (b), movimiento de dedo (c) y toque de pantalla (d).

Reconocimiento 3D

Para solucionar el problema anterior se ha adaptado el algoritmo de reconocimiento de gestos "\$1 recognizer", en 2D para su uso en 3D. \$1 muestrea la figura geométrica rotándola, escalándola y normalizando su longitud antes de compararlo con los gestos almacenados en su base de datos. En una segunda versión del algoritmo, "\$1 Protractor", el trazo se descompone en solamente 16 puntos, en el siguiente paso se normaliza el vector y se rota en torno al ángulo base y finalmente se redimensiona. Ambas versiones del algoritmo pueden ser elegidas desde la interfaz de la aplicación Leap Learn.

Para el reconocimiento de los gestos 3D, se ha optado por separar el gesto realizado en sus imágenes reflejadas en los planos ortogonales XY, YZ y XZ. Ver Figura 4.

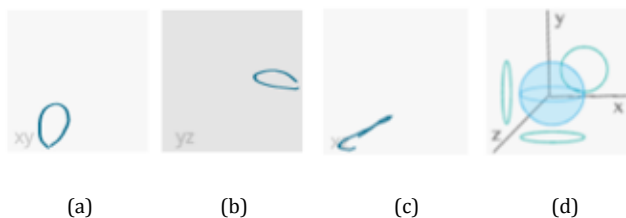


Figura 4, planos ortogonales XY (a), YZ (b) y XZ (c) de una figura geométrica en 3D (d).

Para conseguir el gesto ganador se detecta el plano ortogonal predominante en el que el gesto ha sido realizado y se ejecuta el algoritmo \$1 comparando la entrada con la base de datos de gestos.

Aplicación Leap Learn

En la pantalla principal de la aplicación (Figura 5) se muestran las siguientes partes: 1. control y lista de gestos desde donde crear un nuevo gesto, y visualizar la lista de los existentes, 2. zona donde se realizan los gestos, 3. panel de información de los ejes que están siendo leídos por el algoritmo, 4. controles de los parámetros del algoritmo, 5. elección del vértice

metacarpiano o distal y 6. gráficas separadas en los planos ortogonales: XY, XZ e YZ.

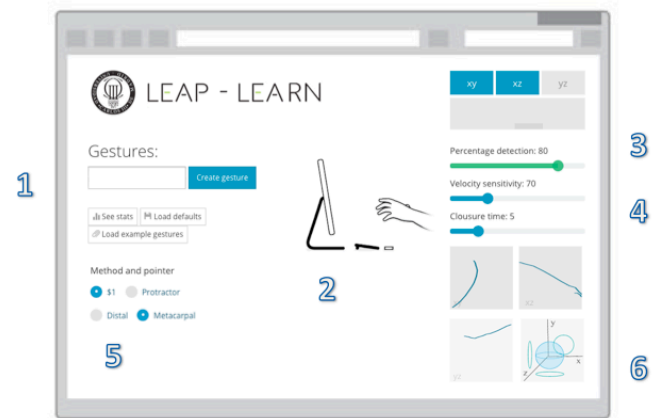


Figura 5, Pantalla principal de la aplicación Leap Learn.

El gesto comienza a generarse cuando la aceleración de la mano es suficiente y se detiene cuando la aceleración decrece y el tiempo de holgura finaliza.

Para crear gestos, es necesario introducir el nombre del mismo en el campo de texto de la zona 1, y pulsar sobre el botón "Crear gesto". un indicador temporal, ver Figura 6, permite durante 3 segundos situar la mano en el campo de visión de Leap Motion, y comenzar el gesto.

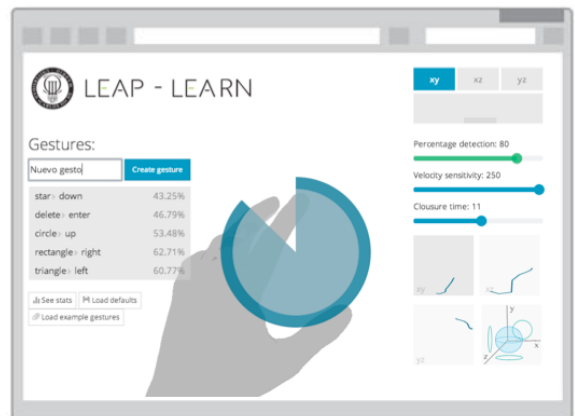


Figura 6, grabación de un gesto.

El gesto no se grabará hasta que se haya leído un gesto completo, con inicio y parada. En el momento que el gesto ha sido registrado, se pide al usuario introducir una tecla del teclado para asignar una acción. Al finalizar, será añadido a la lista de gestos.

La acción que se ejecutará será el gesto de mayor valor que esté por encima del umbral de decisión, definido en el parámetro "decision's percentage".

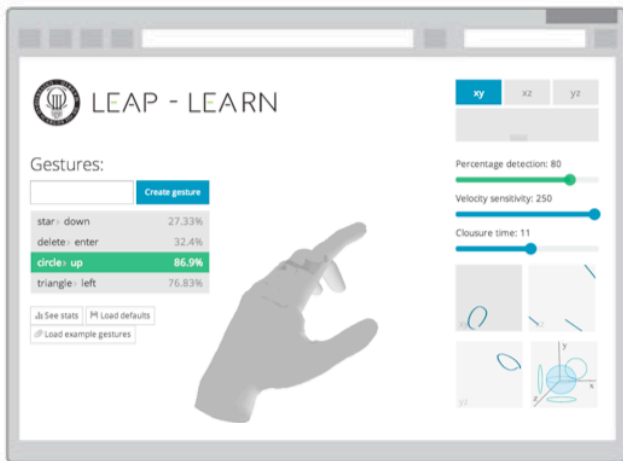


Figura 7, gesto Círculo leído, con un porcentaje ganador del 86,9%.

El resultado del gesto vencedor es resaltado en la lista de gestos como se representa en la Figura 7.

V. EVALUACIÓN DE LA APLICACIÓN

Para medir la usabilidad de “Leap Learn”, una muestra de 6 usuarios [34] comprendidos entre 20 y 40 años, habituados al uso de la informática sin ser programadores., han evaluado la herramienta dentro de un entorno doméstico.

En el experimento realizado ha consistido en la introducción de 3 nuevos gestos por cada uno de los participantes asociando las diferentes acciones de teclado. En un segundo paso, se han repetido los gestos durante varias ocasiones para comprobar la eficacia de los mismos.

La experiencia de usuario experimentada se refleja en las respuestas del cuestionario de la escala SUS, realizado inmediatamente después de terminar el experimento.

La escala SUS es un cuestionario estándar [35] de estilo Likert, con únicas respuestas valoradas entre 1 y 5 (ver Tabla 1), que genera un único número en representación de la medida compuesta de la usabilidad del sistema global. A través de las diferentes preguntas de SUS, se mide la **eficacia** (los usuarios pueden lograr con éxito sus objetivos) **eficiencia** (la cantidad de esfuerzo y recursos que se gastan en la consecución de esos objetivos) y **satisfacción** (experiencia satisfactoria) del sistema global sometido a estudio.

Título de la pregunta		
Completamente en desacuerdo	Valores de 1 a 5	Completamente de acuerdo

Tabla 1, escala Likert.

El listado de preguntas que han respondido los usuarios se muestra a continuación:

- Definición de gestos 3D por usuario final. Una experiencia práctica con Leap Motion.

- Creo que me gustaría visitar con frecuencia este sistema.
- Encontré el sistema innecesariamente complejo.
- Pensé que era fácil utilizar el sistema.
- Creo que necesitaría del apoyo de un experto para utilizar el sistema.
- Encontré las diversas posibilidades del sistema bastante bien integradas.
- Pensé que había demasiada inconsistencia en el sistema.
- Imagino que la mayoría de las personas aprenderían rápidamente a utilizar el sistema.
- Encontré el sistema dificultoso al recorrerlo.
- Me sentí muy confiado en el manejo del sistema.
- Necesito aprender muchas cosas antes de manejarlo en el sistema.

La puntuación de la escala produce un resultado dentro del rango de 1 a 100 que no se debe interpretar como un porcentaje.

De acuerdo al resultado de 500 estudios con la escala SUS [36] se determinó que su valor medio fue de 68 puntos. Tomando como referencia este valor, se puede decir que el nuevo sistema estudiado estará por encima o por debajo de la media establecida. Por otra parte, en complemento a esta definición, se ha establecido una calificación de los diferentes resultados de la escala [37] que establece zonas de aceptación, diferentes grados (de A a F) y adjetivación de los resultados de la escala como se muestra en la Figura 8.

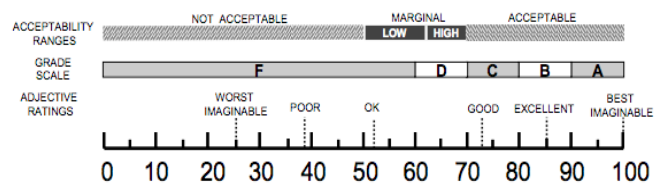


Figura 8, comparación de las calificaciones de adjetivos, aceptabilidad y promedio de graduación de la escala SUS.

Los resultados de la escala SUS obtenidos a partir de la muestra de usuarios que han participado proporcionan una media aritmética de 60,083, puntos como se detalla en la Tabla 2.

Resultados Usuarios SUS	67.5	55.5	60	62.5	52.5	62.5
Media aritmética	60,083					

Tabla 2, resultados de la escala SUS.

El resultado se sitúa entre los valores “suficiente” (ok) y “bien” (good), por debajo de la media establecida (68).

En la Figura 9 se muestran la media de las diferentes puntuaciones para las preguntas del cuestionario SUS.

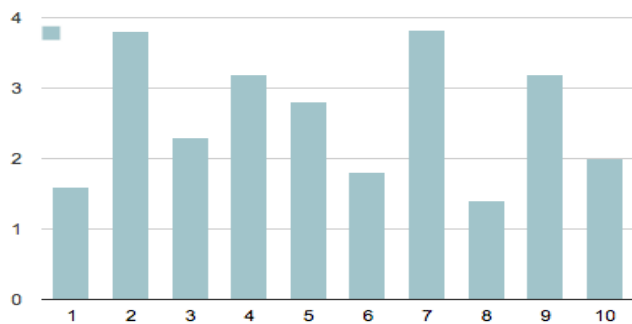


Figura 9, media aritmética para los resultados de cada pregunta de SUS.

El estudio muestra que la aplicación Leap Learn es eficaz, logrando la mayoría de las veces sus objetivos. A nivel de eficiencia, los usuarios necesitan hacer un esfuerzo complementario para realizar de la forma más precisa posible los gestos que fueron grabados anteriormente, acción que repercute directamente en un nivel menor de satisfacción.

Algunas de las sensaciones generales que se recogen en este estudio entre los participantes se desglosan a continuación:

- Dibujar figuras sin una superficie de contacto dificulta la ejecución de las mismas, provocando que en cada intento pueda realizarse un gesto completamente diferente.
- El usuario necesita aprender y recordar el espacio no visible que Leap Motion reconoce, lo que en ocasiones provoca que los gestos no sean del todo naturales al estar condicionados por el rango de visión.
- Leap Motion necesita reconocer los elementos que están en su campo de visión, lo que provoca que desde que el usuario introduce sus manos dentro del campo de visión, hasta que la herramienta los reconoce, se produce un lapso de tiempo donde la herramienta no produce ningún movimiento que puede confundir al usuario.
- El uso extendido de la aplicación produce fatiga o cansancio de los brazos en sesiones largas de uso.

Algunos problemas derivados del diseño de Leap Motion influyen de manera negativa en la experiencia del usuario como se muestran a continuación:

- Se requiere un volumen alto de procesamiento en el cálculo de los datos de entrada que puede ralentizar el sistema en momentos de alta actividad.
- El campo de visión queda reducido a una sola capa. Leap Motion no es capaz de detectar objetos que estén sobrepuestos, por ejemplo, dos manos situadas una encima de la otra. Este fenómeno hace que en algunos gestos realizados se vean bloqueados por la falta de visión de los mismos.

VI. CONCLUSIONES

A pesar de que los usuarios se mostraron interesados en la realización de las pruebas, y finalmente pudieron crear por sí mismos nuevos gestos 3D y asociarlos a distintas acciones del ordenador, la usabilidad de la aplicación “Leap Learn” ha concluido en que la usabilidad medida se sitúa por debajo de la media de la escala SUS.

No obstante, los usuarios han declarado sentir especial interés en ver cómo nuevos paradigmas de interacción por gestos en 3D se puedan incorporar en tareas cotidianas de uso diario.

Futuras líneas de investigación

Debido a que “Leap Learn” solamente reconoce uno de los puntos de la mano para la detección de los gestos, permite que en futuros trabajos de investigación se pueda estudiar la capacidad de realizar los gestos midiendo más puntos o incluso todos los que Leap Motion proporciona.

Del mismo modo, se propone mejorar la eficiencia de los algoritmos de reconocimiento que requiera menos tiempo de cómputo para el reconocimiento de objetos de interacción.

VII. BIBLIOGRAFÍA

- [1] Qureshi, F. Z., & Terzopoulos, D. (2006). Surveillance camera scheduling: A virtual vision approach. *Multimedia systems*, 12(3), 269-283.
- [2] Delaye, A., Sekkal, R., & Anquetil, E. (2011, February). Continuous marking menus for learning cursive pen-based gestures. In *Proceedings of the 16th international conference on Intelligent user interfaces* (pp. 319-322). ACM.
- [3] Gupta, S., Morris, D., Patel, S., & Tan, D. (2012, May). Soundwave: using the doppler effect to sense gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1911-1914). ACM.
- [4] Pu, Q., Gupta, S., Gollakota, S., & Patel, S. (2013, September). Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking* (pp. 27-38). ACM.
- [5] Tang, M. (2011). Recognizing hand gestures with Microsoft's kinect. Palo Alto: Department of Electrical Engineering of Stanford University:[sn].
- [6] Schlömer, T., Poppinga, B., Henze, N., & Boll, S. (2008, February). Gesture recognition with a Wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction* (pp. 11-14). ACM.
- [7] Vanderdonck, J. (2003). Visual design methods in interactive applications. *Chapter, 7*, 187-203.
- [8] Link. Leap Motion Gestures (2015). <https://goo.gl/YpQaec>, 2015.
- [9] Lieberman, H., Paternò, F., Klann, M., & Wulf, V. (2006). *End-user development: An emerging paradigm* (pp. 1-8). Springer Netherlands.
- [10] Soegaard, M., & Dam, R. F. (2012). The Encyclopedia of Human-Computer Interaction. *The Encyclopedia of Human-Computer Interaction*.
- [11] Lieberman, H. (2000). Your wish is my command: Giving users the power to instruct their software. *Morgen Kaufmann*.
- [12] Cook, S. W., & Tanenhaus, M. K. (2009). Embodied communication: Speakers' gestures affect listeners' actions. *Cognition*, 113(1), 98-104.

[13] O'hara, K., Harper, R., Mentis, H., Sellen, A., & Taylor, A. (2013). On the naturalness of touchless: putting the "interaction" back into NUI. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 20(1), 5.

[14] Myers, B. A., Pane, J. F., & Ko, A. (2004). Natural programming languages and environments. *Communications of the ACM*, 47(9), 47-52.

[15] Nardi, B. A. (1993). *A small matter of programming: perspectives on end user computing*. MIT press.

[16] Lieberman, H. (2001). *Your wish is my command: Programming by example*. Morgan Kaufmann.

[17] Myers, B. A. (1988). *Creating user interfaces by demonstration*. Academic Press Professional, Inc.

[18] Cypher, A., & Halbert, D. C. (1993). *Watch what I do: programming by demonstration*. MIT press.

[19] Wobbrock, J. O., Wilson, A. D., & Li, Y. (2007, October). Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology* (pp. 159-168). ACM.

[20] Li, Y. (2010, April). Protractor: a fast and accurate gesture recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2169-2172). ACM.

[21] Lü, H., & Li, Y. (2012, May). Gesture coder: a tool for programming multi-touch gestures by demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2875-2884). ACM.

[22] Kin, K., Hartmann, B., DeRose, T., & Agrawala, M. (2012, May). Proton: multitouch gestures as regular expressions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2885-2894). ACM.

[23] Li, Y., Lu, H., & Zhang, H. (2014). Optimistic programming of touch interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(4), 24.

[24] Lü, H., & Li, Y. (2013, April). Gesture studio: authoring multi-touch interactions through demonstration and declaration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 257-266). ACM.

[25] Yamato, J., Ohya, J., & Ishii, K. (1992, June). Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on* (pp. 379-385). IEEE.

[26] Black, M. J., & Jepson, A. D. (1998). A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *Computer Vision—ECCV'98* (pp. 909-924). Springer Berlin Heidelberg.

[27] Hong, P., Turk, M., & Huang, T. S. (2000). Gesture modeling and recognition using finite state machines. In *Automatic face and gesture recognition, 2000. proceedings. fourth ieee international conference on* (pp. 410-415). IEEE.

[28] Baytaş, M. A., Yemez, Y., & Özcan, O. (2014, October). Hotspotizer: end-user authoring of mid-air gestural interactions. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational* (pp. 677-686). ACM.

[29] Suma, E. A., Krum, D. M., Lange, B., Koenig, S., Rizzo, A., & Bolas, M. (2013). Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit. *Computers & Graphics*, 37(3), 193-201.

[30] Jesús García. (2015) Mac-keyboard node.js module. <https://www.npmjs.org/package/mac-keyboard>.

[31] Código fuente del proyecto Leap Learn. (2015). "<https://github.com/ctwhome/leaplearn>".

[32] Terdiman, D. (2012). Leap Motion: 3D hands-free motion control, unbound. *Cutting Edge, c/ net*. Retrieved.

[33] Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human-Computer Interaction*, 24(6), 574-594.

[34] Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.

[35] Nielsen, J. (1999). *Designing web usability: The practice of simplicity*. New Riders Publishing.

[36] Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3), 114-123.

[37] Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3), 114-123.

AGRADECIMIENTOS

Quiero hacer mención especial en agradecimiento a mis tutores por su constancia e infinita ayuda para realizar este trabajo.

Agradecer a la familia y amigos que me han apoyado en la realización de este máster, en especial a Anna van Duin, y al apoyo obtenido por la comunidad de desarrolladores de Leap Motion.

AUTORES



Jesús García González, nació en Madrid, España, en 1986. Obtuvo el título de Ingeniero Técnico en Informática de Gestión en la Universidad Carlos III de Madrid en 2012. Actualmente cursa el Máster de Ciencia y Tecnología Informática en la misma universidad. Sus líneas de investigación se centran en la interacción persona-ordenador, comprendiendo el estudio, planificación y diseño de interfaces.

TUTORES



Andrea Bellucci, postdoctorado en la Universidad Carlos III de Madrid. Doctor en Ciencias de la computación por la misma universidad trabajando junto a Ignacio Aedo y Paloma Díaz en el departamento del DEI. Trabaja como investigador en la UC3M en las áreas de Desarrollo por usuario final y prototipado rápido de sistemas ubíquos, sistemas de sensores e interacción tangible.



Ignacio Aedo Cuevas, es doctor por la Universidad Politécnica de Madrid y Catedrático de Universidad en la Universidad Carlos III de Madrid. Sus intereses investigadores se centran en los sistemas interactivos para la educación, los sistemas web, las metodologías de desarrollo y los sistemas de información para situaciones de emergencia.