

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Análisis y comparativa de técnicas de apuntado y clic aplicando detección de gestos manuales mediante Leap Motion



Grado en Ingeniería Informática

Trabajo Fin de Grado

Melchor Lafuente Duque

Asier Marzo Perez, Sonia Elizondo Martinez

Pamplona, 19 de enero de 2023

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Agradecimientos

Quiero agradecer a mi director Asier Marzo la apuesta por invertir su tiempo y dedicación en orientarme y ayudarme con este trabajo, sin él no podría haber desarrollado un proyecto tan imaginativo y completo como este. Del mismo modo, quiero agradecer a mi co-directora Sonia Elizondo por ayudarme a organizar mis ideas y por ayudarme con las pruebas de usuarios al igual que Unai J. Fernandez y el resto del equipo de UpnaLab.

Finalmente quisiera agradecer a mis amigos, amigas y familiares por haberme alentado durante las dificultades que trae consigo el paso del tiempo.

Abstract

A gesture input technique has been developed for computers running a *Windows* operating system. This technique allows the user to have bimanual control over the left click of the mouse through pinch gestures.

Subsequently, a user study has been carried out to test the benefits of using this input method in comparison to another equally complex and other more traditional ones, with the aim of knowing how beneficial this new technique is.

Finally, the conclusion has been reached that the system developed in this work could, without a doubt, compete, not only with the mouse and keyboard, but also with existing manual interaction programs.

Keywords: pinch gestures, input technique, manual interaction, point and click, text entry, Leap Motion, user evaluation

Resumen

Se ha desarrollado una técnica de reconocimiento de gestos para ordenadores con sistema operativo *Windows*; la cual permite al usuario tener un control bimanual sobre el clic izquierdo del ratón mediante gestos de pellizco.

Posteriormente, se ha realizado un estudio de usuario comparando este método de captación de la información con otro igual de complejo y otros más tradicionales, con el objetivo de conocer cuán beneficiosa resulta esta nueva técnica.

Finalmente, se ha llegado a la conclusión de que el sistema desarrollado en este trabajo podría, sin lugar a duda, competir, no solo con el ratón y el teclado, sino también con programas de interacción manual ya existentes.

Palabras clave: gestos de pellizco, técnica de entrada, interacción manual, apuntado y clic, entrada de texto, *Leap Motion*, estudio de usuario

Índice

1. Introducción	5
1.1. Estado del arte	5
1.2. Motivación	6
1.3. Objetivo principal	6
2. Sistema desarrollado: <i>Leap Pointer</i>	7
2.1. <i>Hardware: Leap Motion Controller</i>	7
2.2. <i>Software</i>	9
2.2.1. Funcionalidades	10
2.2.2. Composición de la escena en <i>Unity</i>	14
2.2.3. Código	16
3. Otra técnica de interacción manual: <i>Ultraleap's TouchFree</i>	21
3.1. Hardware y ajustes del sistema	21
3.2. Software	23
3.2.1. Funcionalides	23
4. Estudio de Usuario	25
4.1. Participantes	25
4.2. Tarea de apuntado y clic	25
4.3. Tarea de escritura	26
4.4. Procedimiento	27
4.5. Mediciones	27
4.6. Resultados	28
4.6.1. Resultados Objetivos	28
4.6.2. <i>Visualización de los resultados de la tarea de apuntado y clic</i>	30
4.6.3. <i>Visualización de los resultados de la tarea de escritura</i>	36
4.6.4. Resultados Subjetivos	42
4.6.5. Interpretación	42
4.7. Discusión	43
5. Conclusiones	44

6. Líneas Futuras	45
Índice de figuras	46
Referencias y citas bibliográficas	48

1. Introducción

1.1. Estado del arte

Las técnicas de reconocimiento de gestos son parte de un conjunto de tecnologías que tienen como objetivo interpretar gestos humanos a través de algoritmos matemáticos, entendiéndose por gestos humanos toda la gama de movimientos corporales. De esta manera, existen programas que se centran en el seguimiento de la vista, como *OptiKey*, para ayudar a personas con alguna discapacidad física a controlar el ordenador [1]; otros que se centran en el movimiento genérico del cuerpo, como los juegos de la *Kinect*, para mejorar la psicomotricidad de los usuarios [2]; incluso otros que se centran en los gestos realizados por las manos, como es el caso del programa planteado en este trabajo, para explorar una forma higiénica y sin contacto con la que apuntar y hacer clic en el ordenador, sustituyendo al tradicional ratón, teclado o pantalla táctil. Si bien es cierto que existen programas de interacción manual capaces de lograr la hazaña planteada en este trabajo, como es el caso de *Ultraleap's TouchFree*, del que se hablará más adelante, se sabe que no son sistemas cómodos de usar a largo plazo debido al cansancio físico que generan en los brazos. A este efecto se le conoce como “brazos de gorila” y es habitual en este tipo de programas [3]. Por ello, el sistema desarrollado en este proyecto viene a competir con sus antecesores, planteando una manera cómoda de utilizar esta tecnología para lograr los objetivos esperados.

Con todo esto, cabe destacar que esta nueva técnica de apuntado y clic nace de extrapolar este mismo método utilizado por Jamin Hu [4] en un proyecto de realidad aumentada. En este trabajo, se hace uso de las *Oculus Rift* para hacer un seguimiento de las manos del usuario en un entorno virtual. Asimismo, aparece un teclado, también virtual, con el que el usuario puede seleccionar la tecla que desea presionar mediante gestos de pellizco como se observa en la imagen de la figura 1.

Así pues, el trabajo presentado en este documento puede también considerarse como una extrapolación para traer el proyecto de Jamin Hu de la realidad extendida al ordenador, obviando la parte de *text entry* y centrándose en la parte de apuntado y clic.

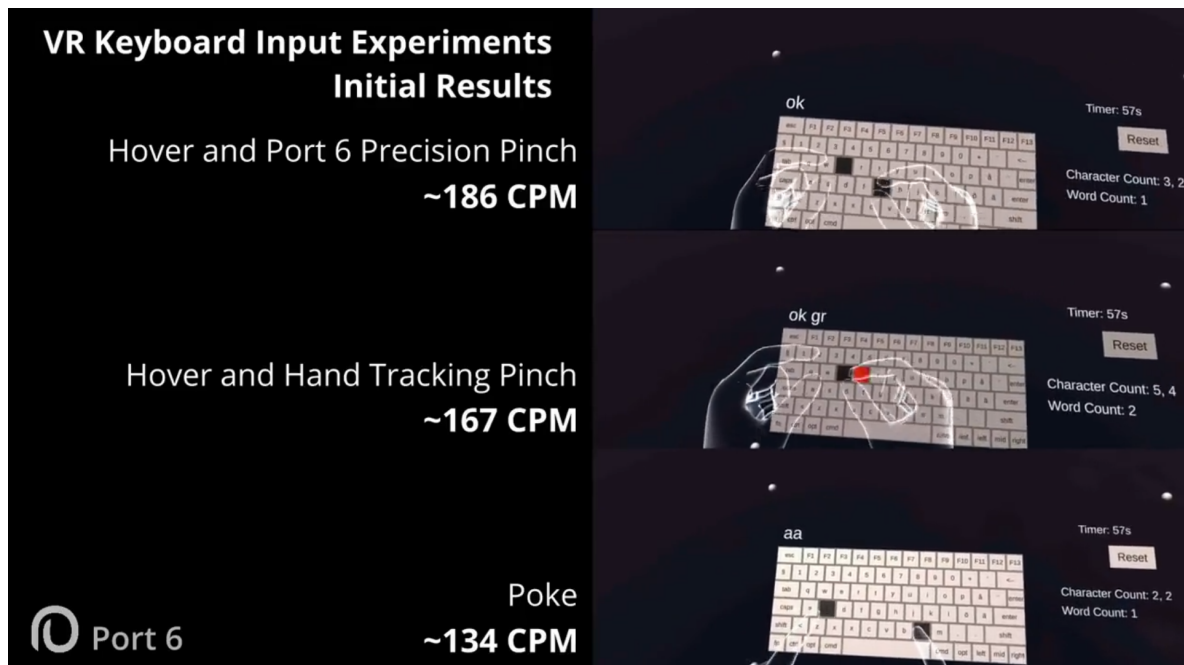


Figura 1: Captura de pantalla tomada del vídeo mostrado en Twitter por Jamin Hu [4].

1.2. Motivación

Tras casi tres años de pandemia debido a la covid-19, la prevención de riesgos sanitarios comienza a ser un elemento fundamental de la sociedad. Por ello, se empiezan a plantear soluciones alternativas a problemas cotidianos que presentaban claros indicios de falta de higiene y salud. Entre estos problemas, se encuentra la utilización de pantallas táctiles comunes como las de los cajeros automáticos o las de compra de entradas.

Partiendo de esta premisa, urge la necesidad de encontrar un método para seguir utilizando sistemas empotrados de forma común de una manera higiénica y sostenible.

1.3. Objetivo principal

El objetivo principal de este trabajo es desarrollar una nueva técnica de reconocimiento de gestos manuales que plantee una posible solución al problema anteriormente mencionado, es decir, una técnica que garantice una vida sana y promueva el bienestar común en todas las edades como define el tercer objetivo de la Agenda 2030 [5].

Para ello, se ha utilizado una *Leap Motion Controller*, con el fin de hacer un seguimiento de las manos del usuario en forma de esqueleto 3D para el motor de videojuegos *Unity*.

Así pues, es en este entorno de desarrollo donde se ha generado un programa capaz de proyectar un *HUD* transparente, el cual muestra un modelo virtual de las manos del usuario sobre un ordenador con sistema operativo *Windows*, proporcionando al usuario un control bimanual sobre el clic izquierdo del ratón mediante gestos de pellizco como se observa en la figura 2.

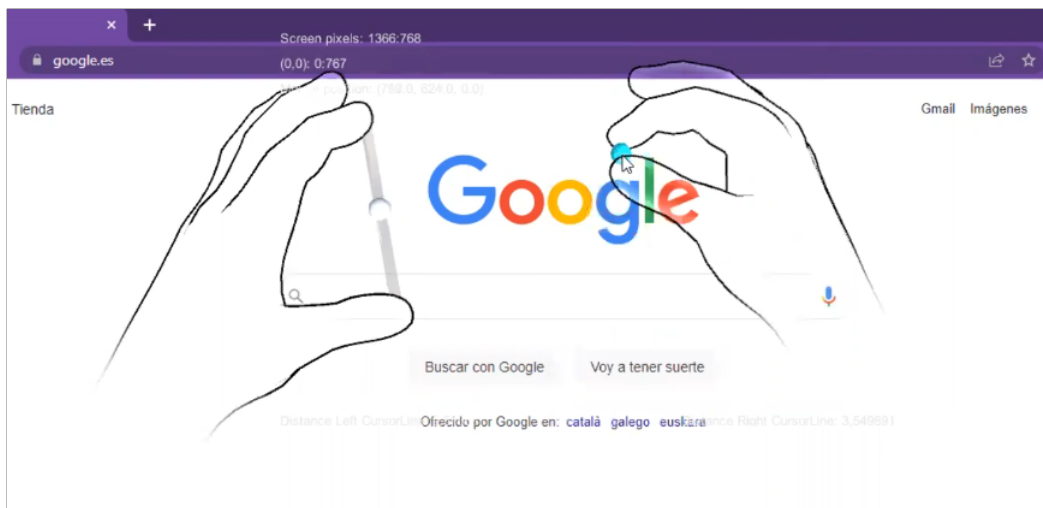


Figura 2: Ejecución del programa *Leap Pointer* en su fase beta.

2. Sistema desarrollado: *Leap Pointer*

2.1. *Hardware: Leap Motion Controller*

La tecnología de *Leap Motion Controller* comenzó a desarrollarse en el año 2008, cuando David Holz todavía estaba desarrollando su doctorado en Matemáticas [6]. Así, junto con su socio Michael Buckwald cofundaron la empresa *Leap Motion* en el año 2010 [7]. Esta consiguió recaudar una financiación inicial de 1.3 millones de dólares en junio de 2011 con inversiones de las empresas de capital de riesgo *Andreessen Horowitz*, *Founders Fund* y *SOSV*, entre otros [8].

Después de operar en silencio desde 2010, *Leap Motion* anunció públicamente su primer producto, originalmente llamado *The Leap*, el 21 de mayo de 2012. La compañía lanzó un programa para desarrolladores de software en octubre de ese mismo año [9] y estableció un plan para distribuir aproximadamente 12000 unidades a los desarrolladores interesados en crear aplicaciones para el dispositivo [10].

En mayo de 2014, *Leap Motion* lanzó su segunda versión de software para desarrolladores en una versión beta pública [11]; y en agosto, la empresa lanzó un modo de seguimiento de realidad virtual, diseñado para proporcionar seguimiento manual mientras el dispositivo está montado en visores de realidad virtual como *Oculus Rift* [12]. Posteriormente, en el año 2016, *Leap Motion* desarrolló un nuevo software para trabajar en un entorno *VR* con el dispositivo *Leap Motion Controller* llamado *Orion* [13].

En el año 2019, la empresa *Ultrahaptics* compró a *Leap Motion*, fusionándose en una nueva empresa de nombre *Ultraleap* [15]. Hoy día, tal y como presumen en su página web oficial[16], han conseguido unir el seguimiento de manos más avanzado del mundo con la única tecnología háptica que crea la sensación del tacto en el aire. De esta forma, *Ultraleap* cuenta con un equipo de más de 150 personas repartidas por todo el mundo, con ubicaciones en Silicon Valley (EE.UU.) y Bristol (Reino Unido). Este equipo incluye expertos líderes mundiales en diseño de interfaz, acústica, aprendizaje automático y visión por computadora.

Tras este desarrollo, el dispositivo *Leap Motion Controller*, abreviado a partir de este momento como *Leap Motion*, ha ido evolucionando hasta llegar a constar de una cámara de infrarrojos con un software integrado de quinta generación llamado *Gemini*, mediante el cual es posible realizar un seguimiento de las manos del usuario con gran precisión. De esta manera, devuelve un esqueleto en 3D de las manos del usuario al *PC* en tiempo real como se observa en la Figura 3.



Figura 3: Ejemplo del funcionamiento de *Leap Motion* [14].

Ventajas de utilizar el *Leap Motion*

Lo primero a destacar es que su función de seguimiento es muy completa, debido a que la latencia con la que obtiene los gestos de las manos es baja y a que captura los movimientos casi a la perfección. Además, los requerimientos mínimos [17] para utilizar el aparato hacen que su usabilidad se extienda entre un amplio número de sistemas.

En cuanto a la documentación [19], está actualizada y permite trabajar con múltiples lenguajes de programación y entornos. Entre ellos se encuentra el motor de videojuegos *Unity*, el cual se ha utilizado para desarrollar el código del programa principal planteado en este proyecto.

Respecto al apartado *hardware* del sistema, tal y como se puede apreciar en la Figura 4, el diseño del artefacto es bastante minimalista y elegante, lo que facilita la portabilidad de la cámara. Además, cabe destacar que este dispositivo es bastante económico y ronda un precio de mercado de 100€ aproximadamente. Por lo tanto, adquirirlo puede resultar muy barato tanto a particulares como a instituciones públicas.



Figura 4: *Leap Motion* deconstruido[18].

Desventajas de utilizar el *Leap Motion*

Para montarlo, hace falta tener entrada USB 2/3 al ordenador. Eso quiere decir que llevar este proyecto a sistemas empotrados requeriría amoldar la parte *hardware* de dichos sistemas para disponer de una entrada en la que conectar el dispositivo. Además, al cabo de un tiempo, la cámara comienza a calentarse, aunque no de manera alarmante.

Otro punto en contra de *Leap Motion* es que la velocidad de captura de las manos dependerá proporcionalmente del procesador de la máquina a la que este conectada. Por tanto, en procesadores de generación más antigua, puede que los programas que utilicen *Leap Motion* no se sientan tan fluidos.

Aún así, estos pequeños detalles no afectan en absoluto a la usabilidad del dispositivo ni a sus aplicaciones y, por tanto, trabajar con esta cámara en este proyecto supone un completo acierto.

2.2. Software

Mediante el sistema *Leap Pointer*, un usuario puede colocarse frente al ordenador en el que hay conectada una cámara *Leap Motion*. Esta cámara de infrarrojos consigue realizar un seguimiento de cada una de las manos, siempre que sean perceptibles, con una latencia de captura de imágenes inferior a dos milisegundos tal y como indican las especificaciones de la cámara *Leap Motion* [20].

A continuación, es el propio sistema operativo *Windows* el encargado de mostrar un modelo en 3D de las manos, siempre que la cámara *Leap Motion* haya realizado un seguimiento de las mismas. Además, cada mano puede mostrarse o no independientemente de si la otra mano lo ha hecho. También, cabe destacar que los modelos de las manos son lo suficientemente transparentes como para poder ver a través de ellas. Asimismo, cada mano tiene un puntero entre sus dedos índice y pulgar para conocer la posición en la que podrá realizar un clic izquierdo de ratón si lo desea mediante un gesto de pellizco.

El color de cada puntero es variable en base a su función. Por defecto, el color de los punteros es blanco. Al principio, el ratón seguirá al puntero de la mano derecha si las dos son perceptibles por la cámara; si no, seguirá a la única mano que sea capaz de reconocer. Sin embargo, si una de las manos acerca sus dedos índice y pulgar hasta que estén lo suficientemente próximos, sin que la otra lo haya hecho antes, esta mano realizará un clic. En ese momento, el puntero de dicha mano cambiará de color blanco a color azul y mantendrá el clic izquierdo de ratón en medio del puntero hasta que los dedos índice y pulgar vuelvan a separarse lo suficiente. Una vez separados los dedos, el color del puntero cambiará de azul a blanco nuevamente, el ratón disparará el evento de soltar el clic izquierdo y comenzará a seguir al puntero de esta mano, ya que es la última mano en haber realizado un clic.

Además, cada mano puede activar y desactivar su puntero mediante un giro de noventa grados sobre su eje Z siempre que no esté realizando un clic. En ese momento, el color del puntero de dicha mano cambiará de blanco a negro y el ratón seguirá a la otra mano, siempre que esta no esté desactivada.

Así pues, todos los requisitos no funcionales quedan reflejados en la Figura 5.

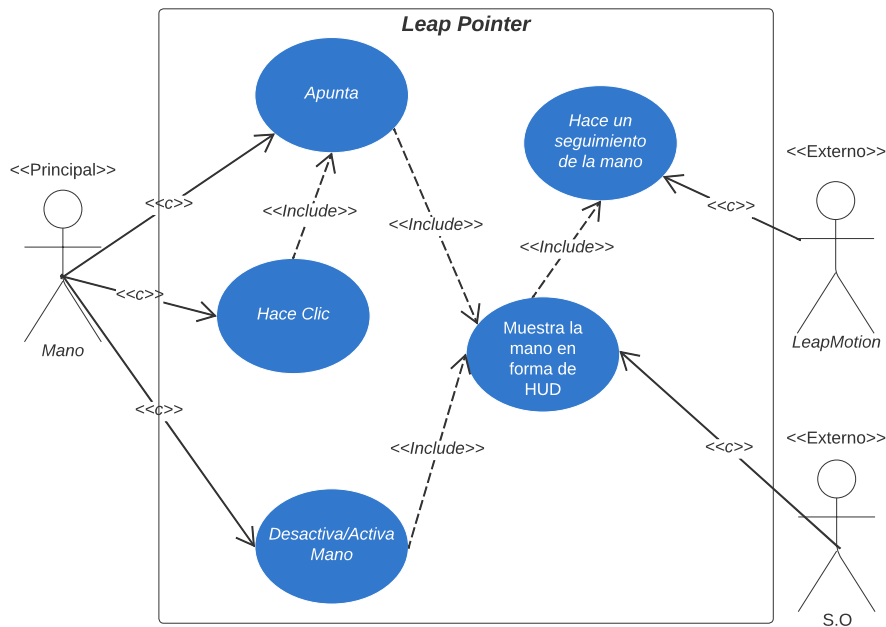


Figura 5: Diagrama de casos de uso del sistema *Leap Pointer*.

2.2.1. Funcionalidades

Resumiendo las cuatro funcionalidades del sistema *Leap Pointer*, el usuario puede:

Apuntar: el color de los punteros es blanco y, al principio, el ratón seguirá al puntero de la mano derecha si las dos son perceptibles por la cámara, si no, seguirá a la única mano que sea capaz de reconocer.

En el ejemplo de la Figura 6, al desplazar la mano derecha del punto 'A' al punto 'B', el ratón sigue al cursor en todo momento.

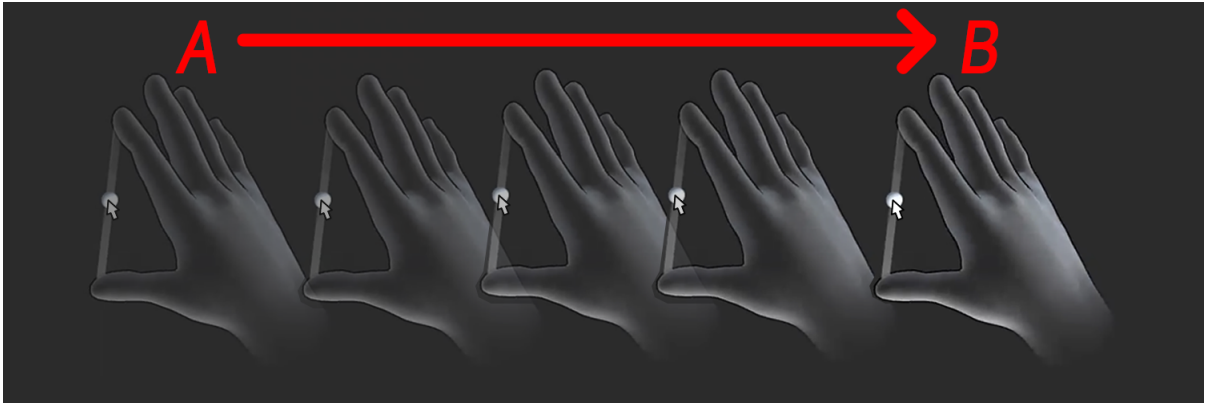


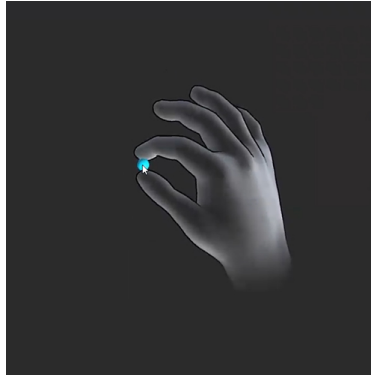
Figura 6: Desplazamiento de la mano derecha.

Clicar: para clicar, el usuario tiene que tener ambas manos en el estado de apuntado, es decir, con los dedos índice y pulgar de cada mano lo suficientemente separados y con cada puntero en color blanco.

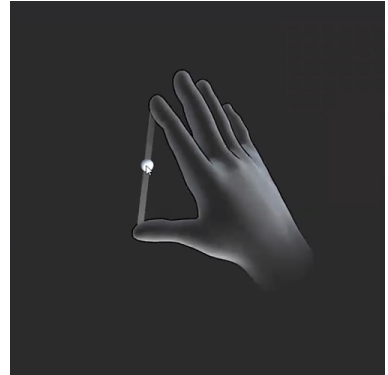
A continuación, con una de las dos manos, deberá entrecerrar sus dedos índice y derecho realizando un gesto de pinza hasta superar un umbral de proximidad haciendo que el puntero cambie de color blanco a azul. En ese momento, allá donde el usuario mueva la mano, se estará ejecutando el evento de pulsar el clic izquierdo del ratón en la posición que ocupe el puntero hasta que los dedos de la mano se extiendan nuevamente. Entonces, se ejecutará el evento de soltar el clic en dicha posición y la mano volverá al estado de apuntado con el añadido de ser la nueva mano a la que sigue el ratón, si es que antes no lo hacía, por haber sido la última mano en haber clicado.



(a) El color del puntero es blanco porque se encuentra en el estado de apuntado.



(b) El color del puntero cambia a azul y se dispara el evento de pulsar el clic izquierdo allá donde esté el puntero.



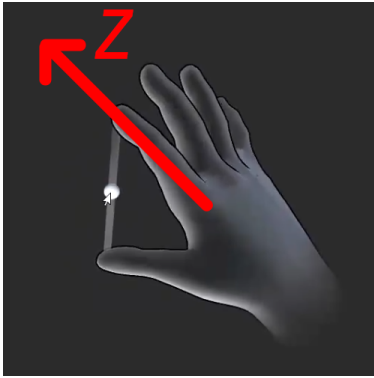
(c) Se dispara el evento de soltar el ratón y la mano vuelve al estado de apuntado.

Figura 7: Mano derecha realizando un clic.

Desactivar la mano: para desactivar alguna de las dos manos, el usuario tiene que tener dicha mano en el estado de apuntado, es decir, con los dedos índice y pulgar de cada mano lo suficientemente separados y con el puntero en color blanco.

A continuación, debe girar su mano noventa grados sobre su eje Z en el sentido que muestra la Figura 8b.

Finalmente, deberá retornar su mano a la posición de origen realizando un giro de noventa grados en el sentido inverso al de antes. En ese preciso instante, el puntero cambiará de color blanco a negro y el ratón seguirá a la otra mano, siempre que esta no esté desactivada.



(a) El color del puntero es blanco porque se encuentra en el estado de apuntado.



(b) La mano realiza un giro de noventa grados sobre su eje Z.



(c) La mano realiza un giro de noventa grados sobre su eje Z en sentido contrario al anterior volviendo a su posición original con el puntero ahora de color negro. El ratón sigue ahora a la mano izquierda siempre que se encuentre activa y sea perceptible.

Figura 8: Proceso para desactivar las funcionalidades de la mano derecha.

Activar la mano: para activar alguna de las dos manos, el usuario tiene que tener dicha mano desactivada, es decir, con el puntero en color negro.

A continuación, debe girar su mano noventa grados sobre su eje Z en el sentido que muestra la subfigura 9b.

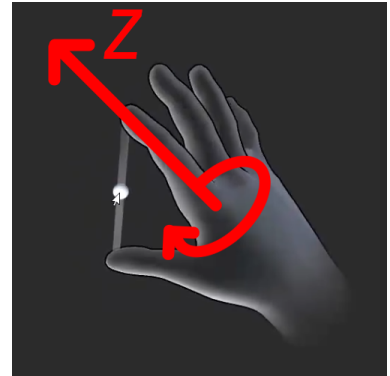
Finalmente, deberá retornar su mano a la posición de origen realizando un giro de noventa grados en el sentido inverso al de antes. En ese preciso instante, el puntero cambiará de color negro a blanco y el ratón seguirá a esta mano siempre que fuera la última en haber realizado un clic o fuera la única mano perceptible.



(a) El color del puntero es negro porque la mano se encuentra desactivada.



(b) La mano realiza un giro de noventa grados sobre su eje Z.



(c) La mano realiza un giro de noventa grados sobre su eje Z en sentido contrario al anterior volviendo a su posición original con el puntero ahora de color blanco. El ratón seguirá a esta mano siempre que haya sido la última en realizar un clic o sea la única mano perceptible.

Figura 9: Proceso para activar las funcionalidades de la mano derecha.

2.2.2. Composición de la escena en *Unity*

Para poder trabajar con el dispositivo *Leap Motion* en *Unity*, lo primero es importar las dependencias necesarias en el entorno de desarrollo. Para ello, la página web oficial de *Ultraleap*[21] ofrece dos métodos:

1. Instalar las dependencias mediante el panel de ajustes de proyecto de *Unity*.
2. Descargar los modelos 3D junto con los códigos necesarios para su correcto funcionamiento en *Unity* a modo de paquete y arrastrarlos manualmente al proyecto donde se desea trabajar con estos.

Habiendo elegido el segundo método para importar el paquete de *Leap Motion* a *Unity*, el segundo paso consiste en montar la escena. Para ello, se ha de arrastrar el controlador al proyecto, el cual, no es más que la representación de la posición que toma el *Leap Motion* en la realidad. De este modo, el modelo del controlador permite elegir qué posición tomará la cámara *Leap Motion* en la realidad de entre las siguientes cuatro opciones:

1. *Head Mounted A*
2. *Head Mounted B*
3. *Desktop mode A*
4. *Screentop*

Ahora bien, de estas cuatro opciones caben destacar única y exclusivamente las dos últimas, ya que *Head Mounted A* y *Head Mounted B* se utilizan para trabajar con las gafas *Oculus Rift* o similares, cosa que el sistema *Leap Pointer* no utiliza.

- *Desktop mode A*: este modo está pensado para que la cámara mire hacia arriba esperando ver las palmas de las manos tal y como se muestra en la Figura 10a.
- *Screentop*: este otro modo está pensado para que la cámara mire hacia abajo. Situándose por encima de las manos del usuario y, por tanto, esperando ver el dorso de las manos tal y como se muestra en la Figura 10b.

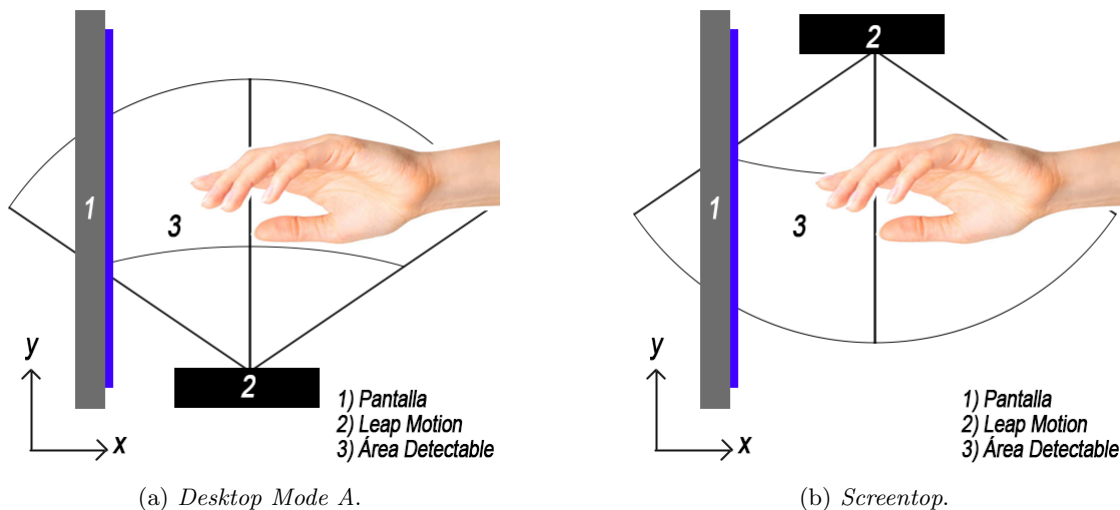


Figura 10: Modos para posicionar la *Leap Motion*.

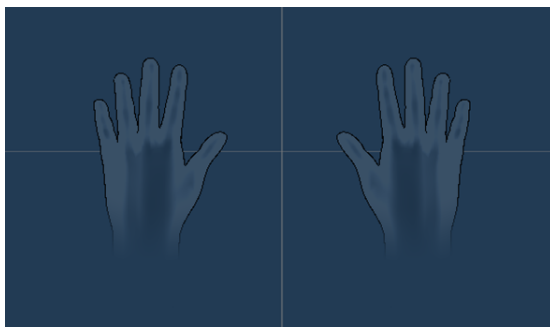
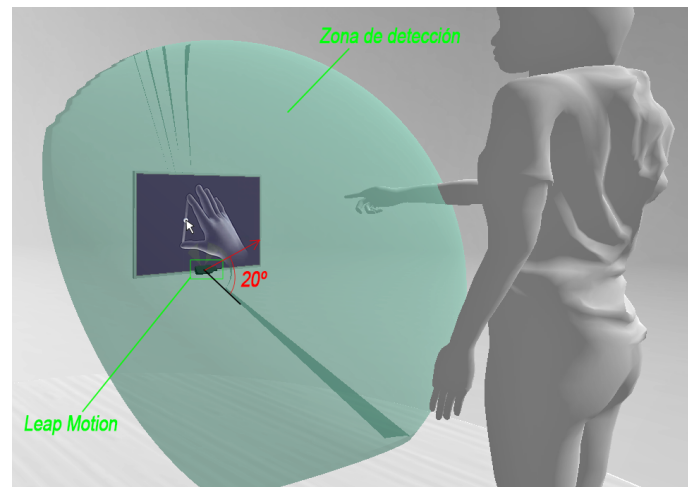
Para el programa interactivo desarrollado en este trabajo, la *Leap Motion*, se ha situado debajo de la pantalla del ordenador con una inclinación aproximada de 20° , como se muestra en la Figura 11.

Por lo tanto, el modo seleccionado para indicarle al controlador cómo debe trabajar es el *Desktop Mode A*, ya que la cámara apunta desde abajo hacia arriba, esperando encontrarse con las palmas del usuario.

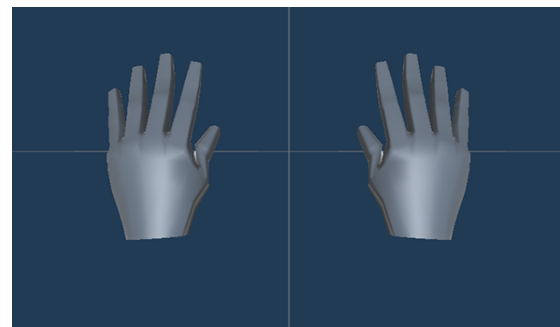
Por último, queda elegir los modelos 3D que representarán las manos del usuario. Entre los posibles modelos se deberá escoger aquel cuyo contorno sea fácil de visualizar sin que impida ver lo que hay más allá, pues estas manos deberán visualizarse como un *HUD* transparente.

Así pues, el paquete extraído de la página web oficial de *Ultraleap*[21] ofrece los siguientes modelos de manos:

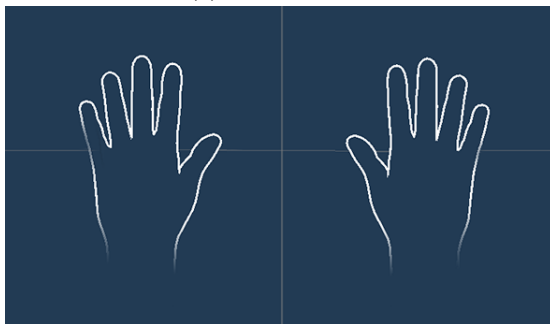
Figura 11: Representación de cómo se coloca la *Leap Motion*



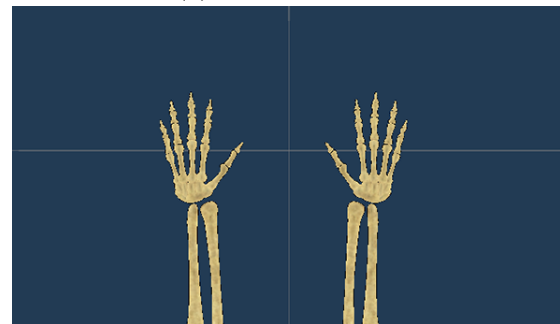
(a) *Ghost Hands*.



(b) *Low Poly Hands*.



(c) *Outline Hands*.



(d) *Skeleton Hands*.

Figura 12: *Prefabs* de manos para *Unity*.

Entre todos estos se ha escogido el modelo de la Figura 12a, ya que su opacidad es transparente pero su contorno es oscuro, permitiendo que se visualicen muy fácilmente en un ordenador con sistema operativo *Windows*.

2.2.3. Código

Para entender el funcionamiento del código implementado, es importante conocer cómo se ha modelado el problema. Para ello, nótese que los elementos que interaccionan entre sí dentro del proyecto de *Unity* son *Game Objects*. Es decir, objetos formados por clases e instancias de clases. Es por ello que en los siguientes esquemas se han modelado los *Game Objects* que toman parte en el proyecto como **componentes**.

Así, la Figura 14 muestra una abstracción muy resumida de los *Game Objects* que hay en el programa *Leap Pointer* y cómo están relacionados entre sí, obviando la *Main Camera* así como las luces de la escena y centrándose en el resto de *Game Objects*. Es decir, este esquema se centra en la relación que existe entre las manos y el controlador de la *Leap Motion*.

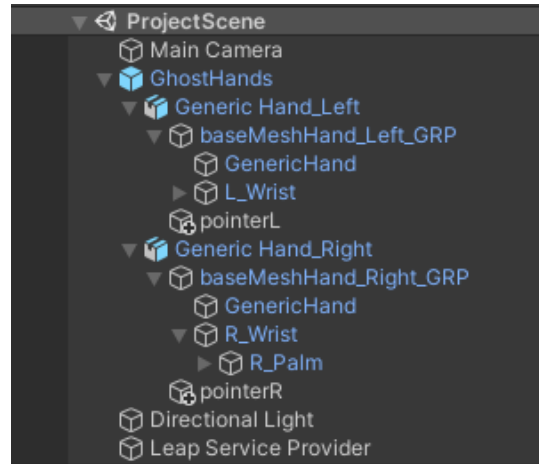


Figura 13: Composición de la escena en *Unity*.

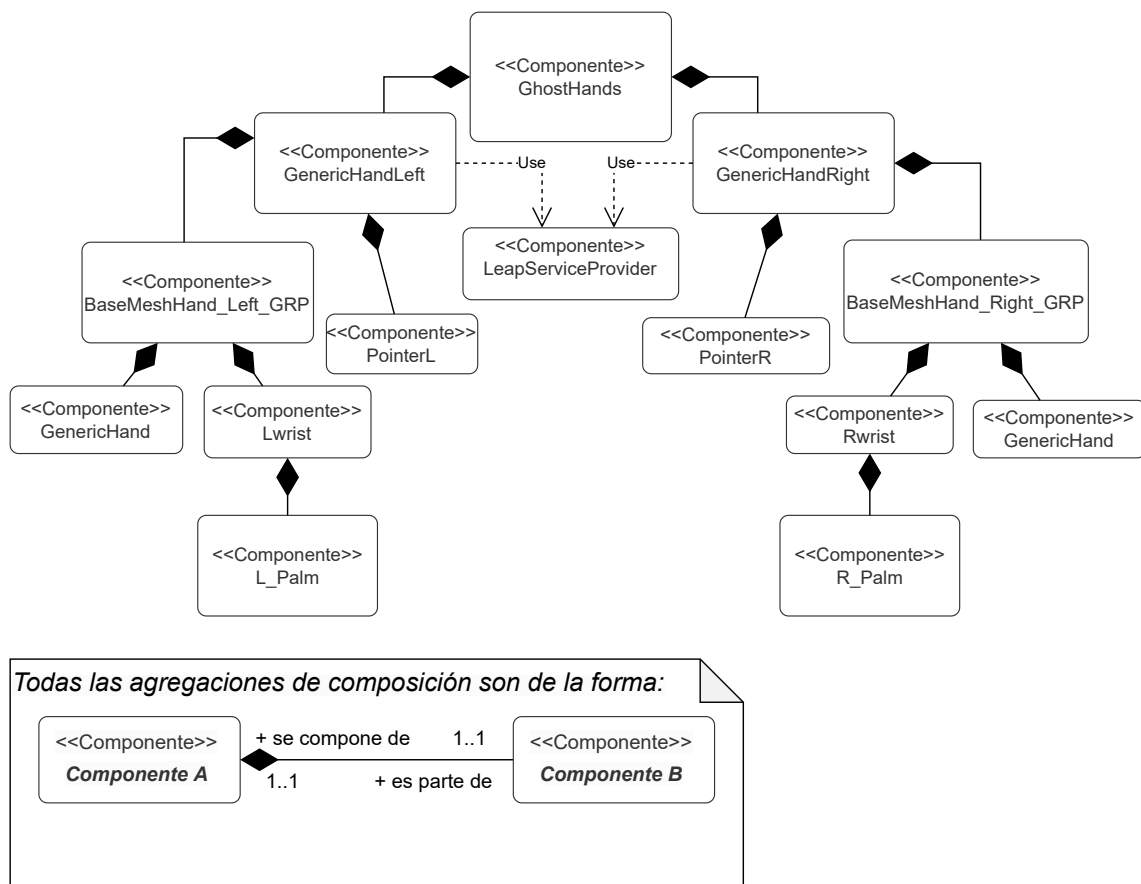


Figura 14: Diagrama de clases del sistema *Leap Pointer* simplificado.

A continuación se muestra el diagrama de componentes que forman al *Game Object* L_Palm:

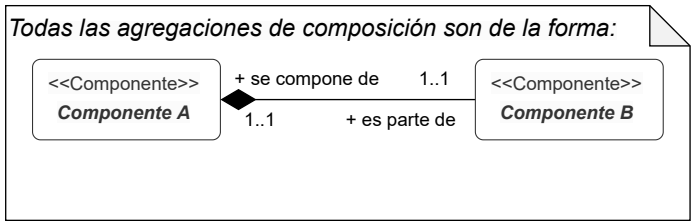
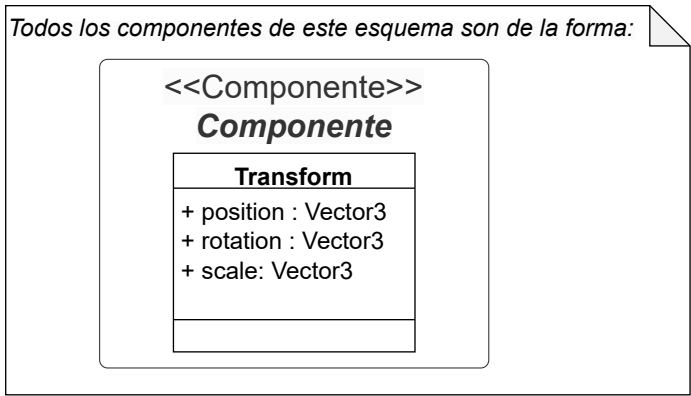
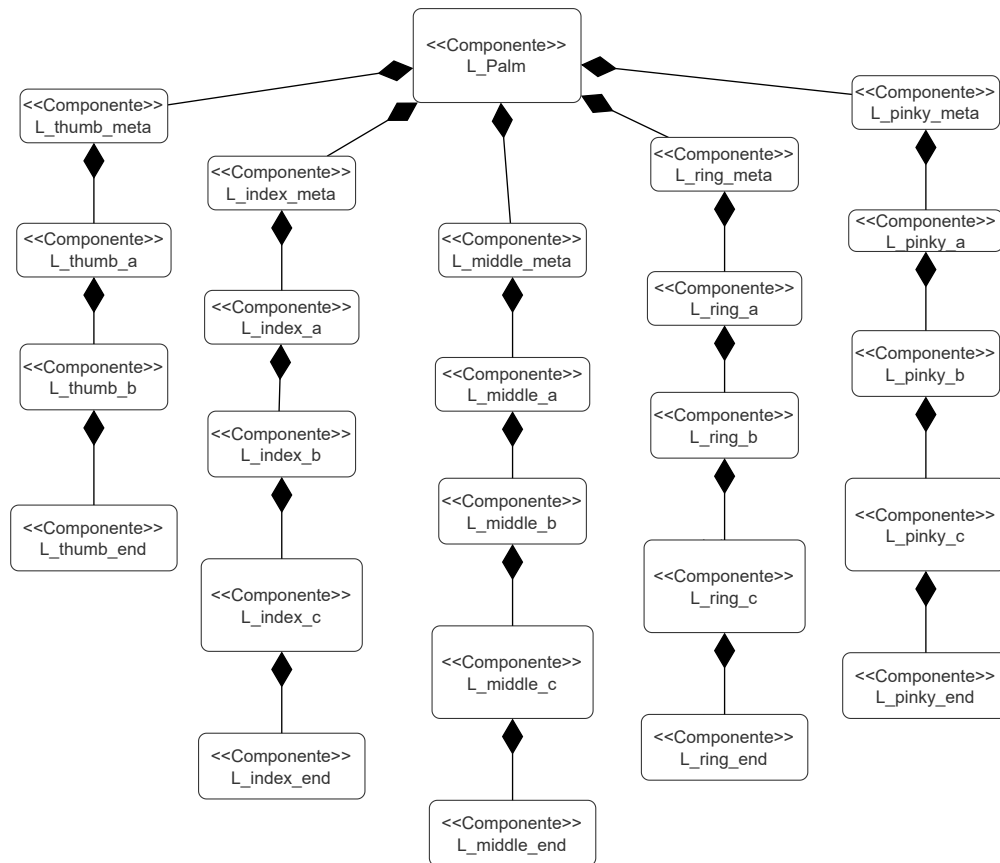


Figura 15: Diagrama de clases de la palma izquierda.

A continuación se muestra el diagrama de componentes que forman al *Game Object R_Palm*:

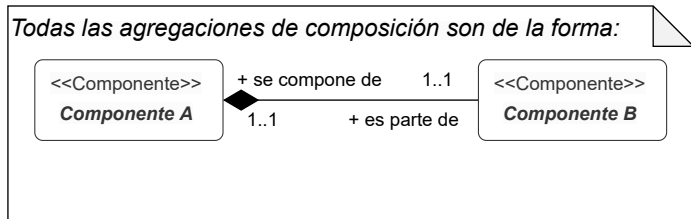
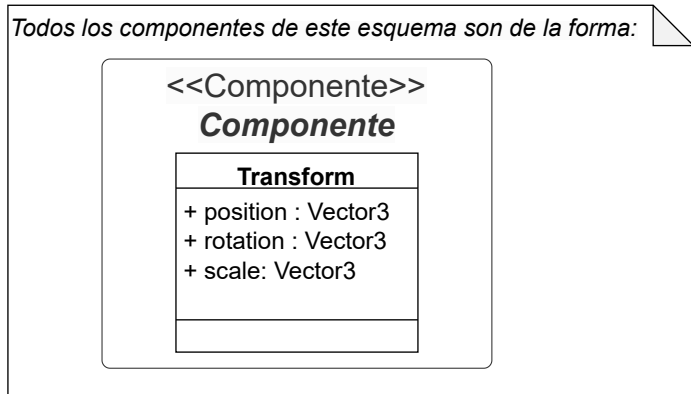
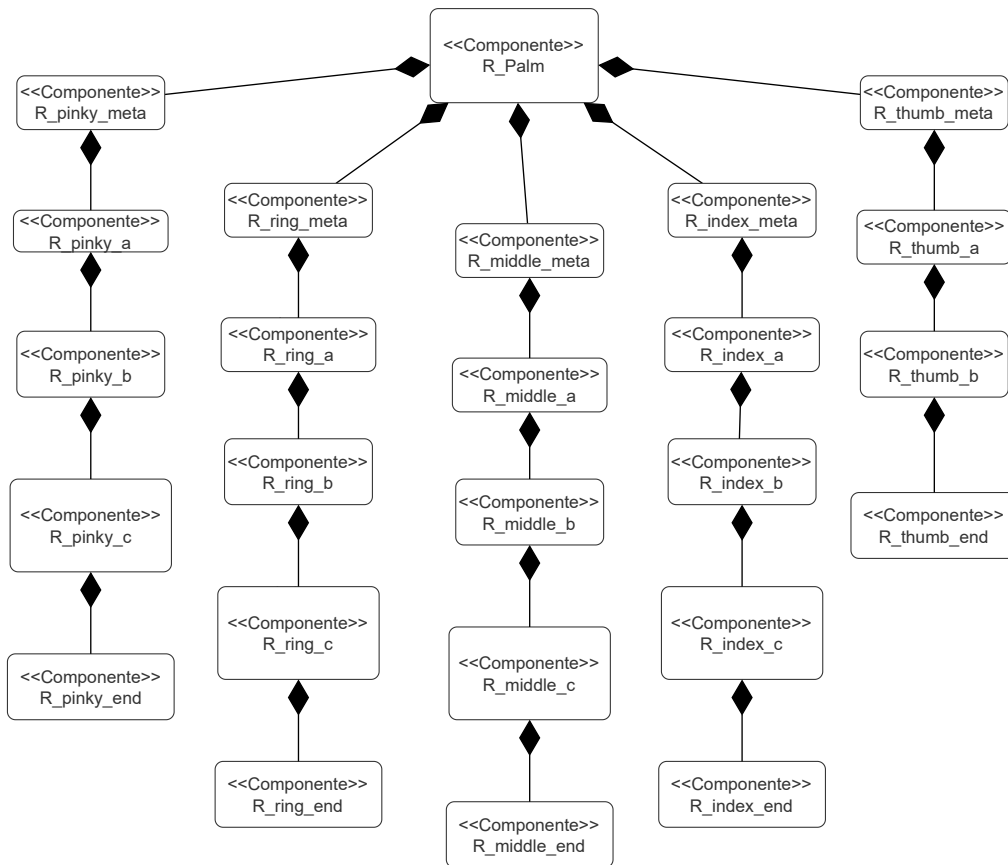


Figura 16: Diagrama de clases de la palma derecha.

Si bien el primer esquema mostraba una abstracción simplificada del problema, la Figura 17 muestra un diagrama de clases en el que se detalla de qué está formado cada componente:

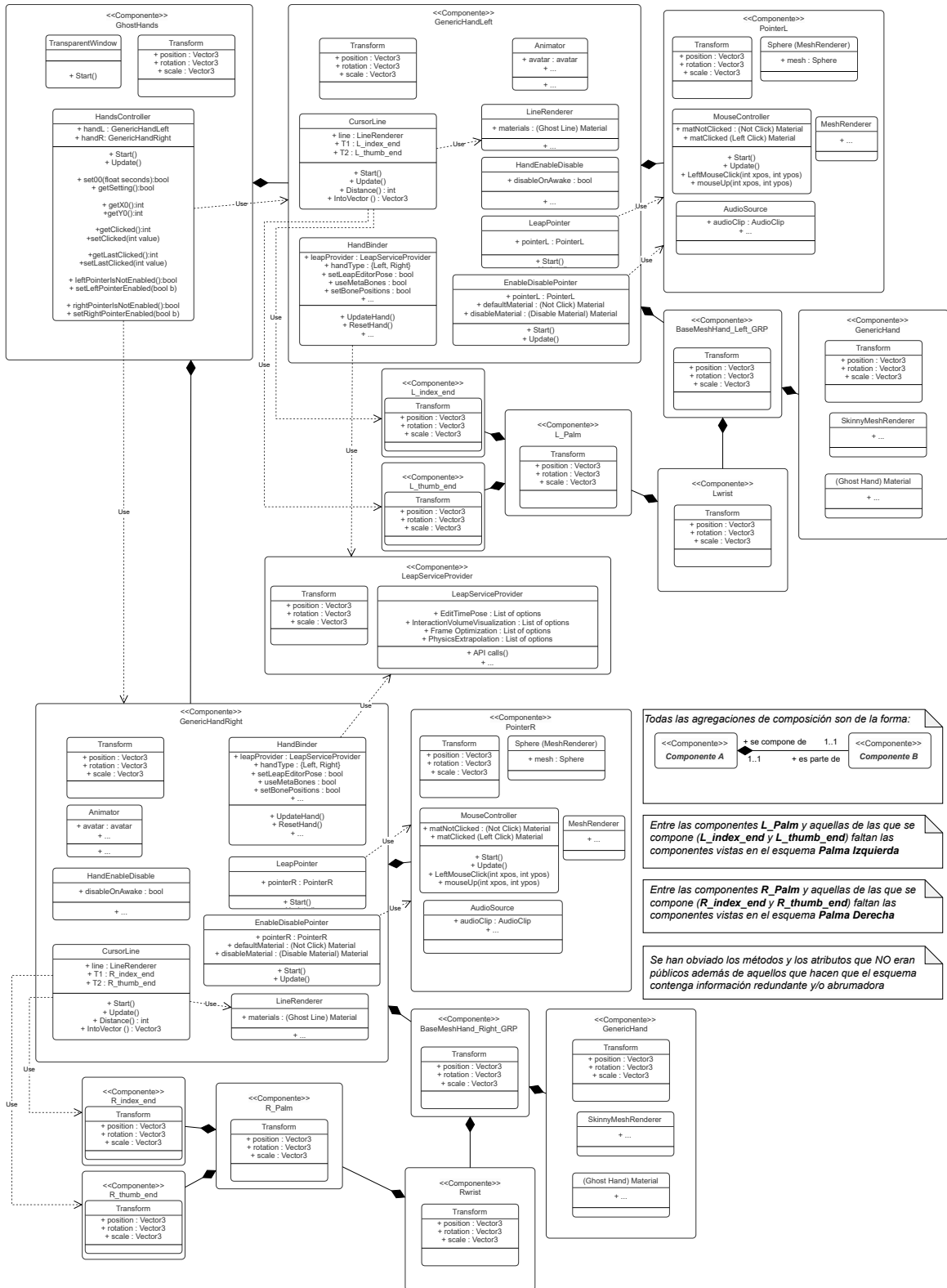


Figura 17: Diagrama de clases del sistema Leap Pointer completo.

De entre todas las clases que existen en el diagrama anterior, merecen especial mención, única y exclusivamente, los *scripts* que repercuten en la ejecución del sistema *Leap Pointer*.

Tomando como referencia la Figura 18, y siguiendo el desglose de los componentes de *GhostHands* de arriba a abajo, se tiene que:

En primer lugar, al *Game Object GhostHands* se le han asignado dos *scripts*:

1. **Transparent Window:** es el código encargado de eliminar el fondo de la escena de *Unity* y ejecutar el programa como un proceso por encima de la interfaz de *Windows*. De esta manera, se consigue dar un efecto de *HUD* al sistema.
2. **Hands Controller:** es el código encargado de ajustar el sistema de coordenadas del cual dependerá el ratón. Gracias a este código, el programa es escalable. Es decir, puede auto-ajustarse a pantallas de diferentes resoluciones a la perfección. Además, este código es el encargado de saber qué mano está haciendo o puede hacer clic en todo momento.

En segundo lugar, a los *Game Objects Generic Hand_Left* y *Generic Hand_Right* se les ha asignado los *scripts*:

1. **Hand Binder:** es el código encargado de realizar un seguimiento de las manos del usuario para posteriormente traducir los movimientos que estas realizan a modo de modelo 3D.
2. **Hand Enable Disable:** es el código encargado de detectar si alguna de las manos del usuario es perceptible por la cámara *Leap Motion* o no. En caso de serlo, activa el modelo 3D de la mano correspondiente. Si no, desactiva el modelo inhabilitando el resto de funciones que el modelo de la mano puede realizar.
3. **Cursor Line:** es el código encargado de generar una línea de tipo *Line Renderer* escalable entre los dedos índice y pulgar de cada mano. Para ello, necesita como parámetros de entrada, a parte de la línea de tipo *Line Renderer*, las componentes *L_index_end* y *L_thumb_end* en el caso de la mano izquierda y *R_index_end* y *R_thumb_end* en el caso de la mano derecha para indicar las posiciones de las puntas de los dedos índice y pulgar de cada mano respectivamente.
4. **Leap Pointer:** es el código encargado de mantener el *pointerL* o *pointerR* (dependiendo de la mano que contenga el código) en medio de cada línea generada por el *script Cursor Line*.
5. **Enable Disable Pointer:** es el código encargado de acceder a las características de *Cursor Line* y conocer si el usuario ha girado la mano sobre su eje Z en función de las posiciones que toma la línea que genera *Cursor Line*. En caso de que el usuario haya realizado un giro de muñeca, desactivará el puntero (*pointerL/pointerR*) correspondiente a la mano a la que pertenezca el código.

Finalmente, a los *Game Objects PointerL* y *PointerR* se les ha asignado el *script*:

1. **Mouse Controller:** es el código encargado de gestionar la posición del ratón en cada momento, así como de realizar el clic y soltar el clic izquierdo del ratón. Para conocer si la mano contenedora del código es la encargada de actualizar la posición del ratón o hacer clic, accede a las características de la clase *HandController* de la componente *GhostHands*.



Figura 18: Desglose del *Game Object GhostHands* hasta sus palmas.

3. Otra técnica de interacción manual: *Ultraleap's TouchFree*

Esta técnica de interacción manual ha sido desarrollada por la empresa con el mismo nombre: *Ultraleap*. Esta empresa es, además, la encargada del desarrollo y el mantenimiento de la cámara *Leap Motion*.

3.1. Hardware y ajustes del sistema

Así pues, respecto al hardware necesario para utilizar su programa interactivo es obvio que utiliza una cámara *Leap Motion* y, de igual manera que lo permitían los ajustes del controlador de la cámara en *Unity*, los ajustes del programa *TouchFree* también permiten situar la cámara *Leap Motion* en diferentes posiciones como lo muestra la figura 19.

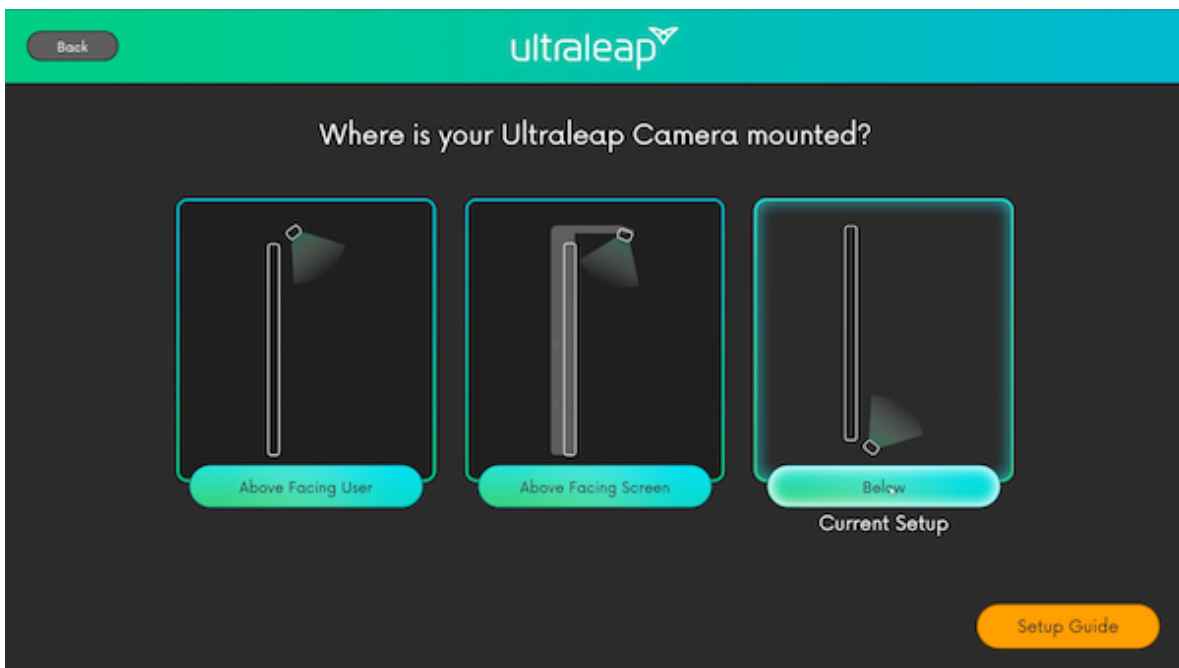


Figura 19: Panel de ajustes de *TouchFree* para indicar la posición que toma la cámara *Leap Motion*[22].

En los estudios de usuario realizados en el siguiente capítulo se situó la cámara de igual manera que lo hacía el sistema *Leap Pointer*, colocando la cámara debajo de la pantalla del ordenador, con una inclinación aproximada de veinte grados y mirando hacia arriba. Por lo tanto, la opción seleccionada para trabajar con el programa *TouchFree* es la opción *Below* (la tercera opción), mostrada en la figura anterior.

Otro de los ajustes que merece especial atención es el hecho de seleccionar, o mejor dicho, deseleccionar la opción *Scroll and drag* que muestra la figura 20. Por defecto, esta opción viene habilitada y permite al usuario tener la capacidad de arrastrar el clic. Sin embargo, al tener esta opción habilitada, el clic no resulta tan fiable. Por tanto, se ha desactivado esta opción para poder comparar el sistema con *Leap Pointer* partiendo de condiciones más parecidas entre sí.

Por último, entre las opciones que ofrece *Tracking Position* de la figura 20, se ha escogido la opción que se llama *Index fingertip*. De este modo, el sistema prioriza los gestos realizados por el dedo índice del usuario que utiliza para apuntar y hacer clic frente al resto de sus dedos. El resto de opciones se han mantenido con sus valores por defecto.

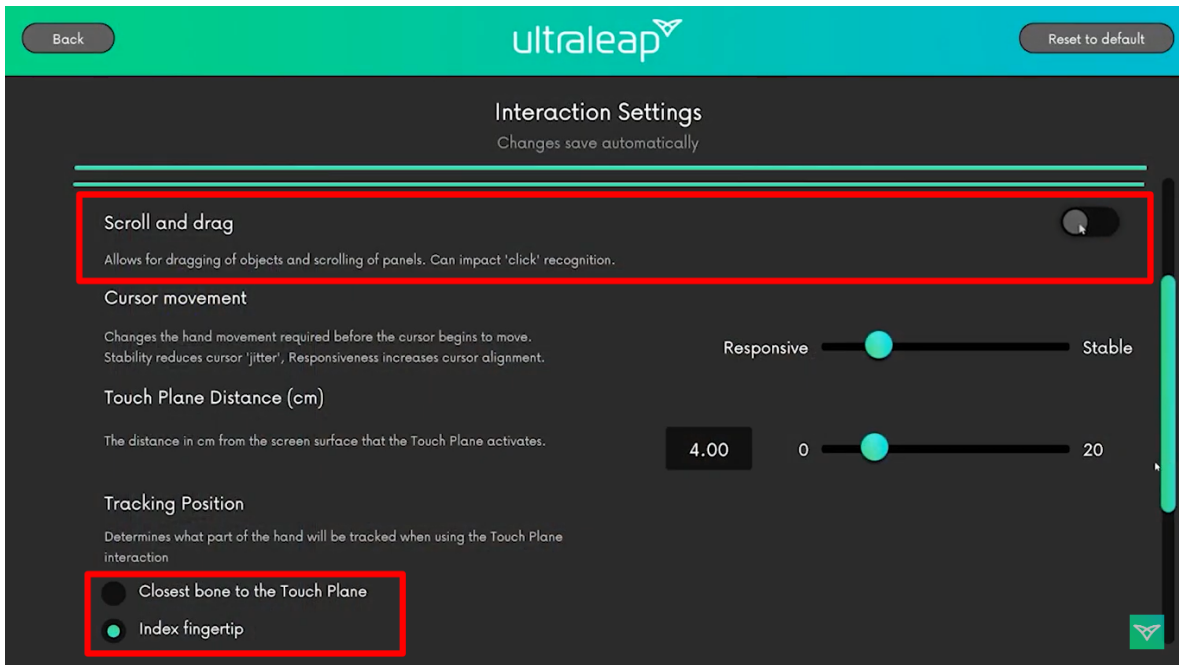


Figura 20: Panel de ajustes interactivos de *TouchFree*.

3.2. Software

Mediante el sistema *TouchFree*, un usuario puede colocarse frente al ordenador en el que hay conectada una cámara *Leap Motion*. Esta cámara de infrarrojos consigue realizar un seguimiento de alguna de las manos del usuario, siempre que sea perceptible, con una latencia de captura de imágenes inferior a dos milisegundos tal y como indican las especificaciones de la cámara *Leap Motion* [20]. Nótese que, al contrario que el sistema *Leap Pointer*, esta técnica no permite la captación bimanual. Es decir, el usuario debe decidir con cuál de sus dos manos interactuará con el sistema.

A continuación, el usuario deberá extender su mano en posición de señalar, con el dedo índice extendido y el resto recogidos, hasta que la cámara *Leap Motion* capte dónde se encuentra su dedo índice. En ese momento, el sistema proyectará un punto en el ordenador indicando la posición a la que apunta el usuario con su dedo.

Si el usuario acerca más y más el dedo índice a la pantalla, hasta llegar a superar un umbral a cuatro centímetros de distancia de la *Leap Motion*, el puntero mostrado en la pantalla se clavará en dicha posición y accionará el evento de mantener pulsado el clic izquierdo de ratón. Cuando el usuario aleje de nuevo el dedo índice, sobrepasando el umbral anterior, el puntero accionará el evento de soltar el clic izquierdo del ratón y volverá al estado de apuntado.

3.2.1. Funcionalidades

Resumiendo las dos funcionalidades del sistema *TouchFree*, el usuario puede:

Apuntar: el color del puntero es blanco y se proyectará en el lugar donde este apuntando el usuario con el dedo índice de alguna de sus dos manos. Nótese que el programa no permite la interacción bimanual y, por lo tanto, el puntero seguirá al dedo índice de la mano que más cerca se encuentre de la cámara.

En el ejemplo de la figura 21, al desplazar la mano derecha del punto 'A' al punto 'B', el puntero también se desplaza acorde a este movimiento.

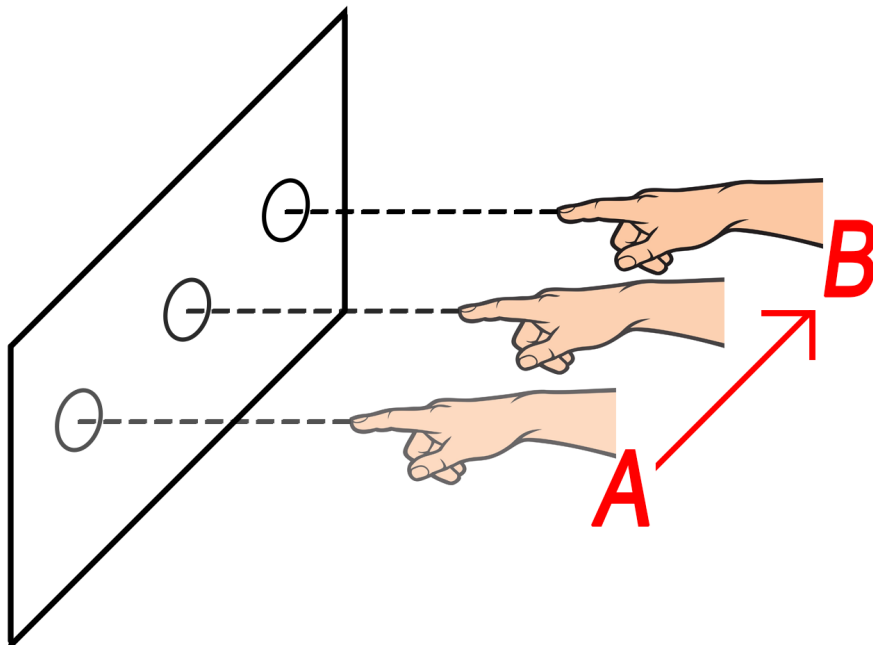


Figura 21: Desplazamiento de la mano derecha.

Clicar: para clicar, el usuario tiene que tener alguna de sus manos en el estado de apuntado, es decir, con su mano en posición de señalar con el dedo índice y lo suficientemente lejos como para no sobrepasar el umbral interactivo, pero lo suficientemente cerca para que la cámara capte sus movimientos. El puntero aparecerá proyectado en color blanco en la posición que este apuntando el usuario.

A continuación, deberá acercar la mano con la que está apuntando hasta superar un umbral de proximidad, a cuatro centímetros de la cámara, haciendo que el puntero se clave en la posición donde apunta y cambie de color blanco a azul. Entonces, se estará ejecutando el evento de pulsar el clic izquierdo del ratón en la posición que ocupa el puntero hasta que la mano se aleje del umbral interactivo nuevamente. En ese momento, se ejecutará el evento de soltar el clic en dicha posición y la mano volverá al estado de apuntado siempre que sea perceptible por la cámara.

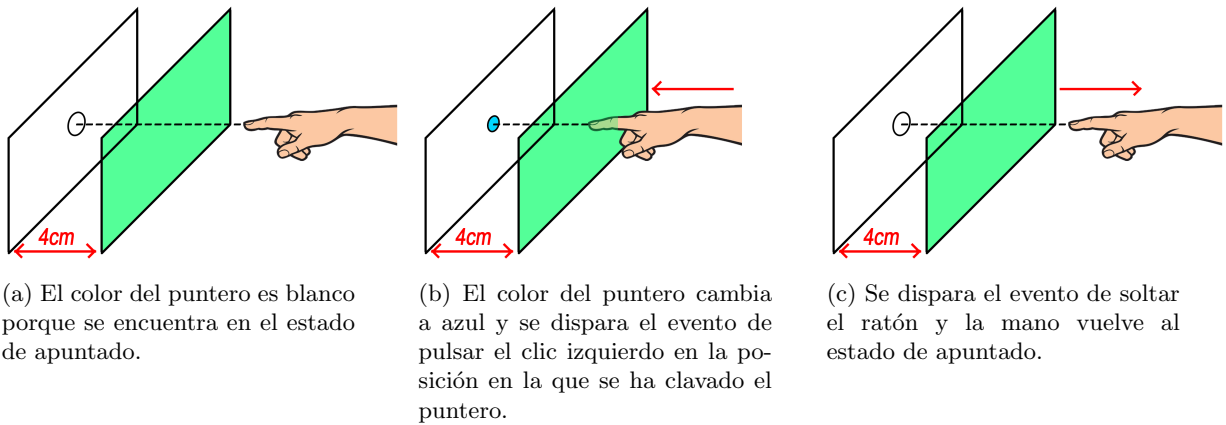


Figura 22: Mano derecha realizando un clic.

4. Estudio de Usuario

Una vez presentados tanto el sistema *Leap Pointer* como el desarrollado por *Ultraleap*, se ha sometido a un número de usuarios a realizar dos tareas, con el fin de contrastar estas dos técnicas de interacción manual con el ratón y el teclado respectivamente. Dichas tareas son:

- **Tarea de apuntado y clic:** en esta tarea se ha comprobado cuán interesante resulta utilizar las diferentes técnicas interactivas para hacer clic en unos puntos específicos de la pantalla.
- **Tarea de escritura:** en esta tarea se ha comprobado cuán interesante resulta utilizar las diferentes técnicas interactivas para clicar en un teclado virtual, y así, escribir. De este modo, no solo se comprueba el acierto obtenido utilizando las diferentes técnicas, sino que también se le suma la velocidad de escribir con *Leap Pointer* o *TouchFree* frente a escribir con teclado, además de la posible frustración o satisfacción que les supone a los usuarios aplicar las técnicas interactivas para solventar una tarea con un fin determinado.

Ambas tareas se han desarrollado como programas en *Processing*, *Java*, donde cada programa muestra una zona interactiva de color blanco con forma cuadrada en medio de la pantalla y, el resto del espacio que ocupa la pantalla, se ha rellenado con color gris, delimitando de esta manera la zona no interactiva. Además, antes de comenzar cada ejercicio, aparece un mensaje con letras en verde que dice 'Presionar para empezar' y, cuando el usuario hace clic sobre cualquier parte de la zona interactiva, la tarea correspondiente da comienzo.

4.1. Participantes

Al querer contrastar las técnicas de *Leap Pointer* con la de *Ultraleap* y el ratón/teclado (dependiendo de la tarea), es decir, al querer contrastar tres técnicas, se necesitan múltiplos de $3! = 3 \times 2 \times 1 = 6$ participantes. De este modo, se han testeado un total de 18 participantes, con edades de entre 19 a 50 años de edad (media = 25.33, desviación estándar = 7.2) y distribuidos de manera cuasi-equivalente según su sexo (7 mujeres y 11 hombres). Algunos de los participantes son personal del departamento de Estadística, Informática y Matemáticas de la UPNA. En concreto, 8 de ellos; 6 de los cuales se han desenvuelto con gran soltura a la hora de realizar las tareas asignadas. El resto de participantes también se ha desenvuelto correctamente llevando a cabo las tareas designadas. Asimismo, ninguno de los participantes ha presentado problemas de psicomotricidad, reflejos o vista, ni ha contado con ningún tipo de malformación física aparente en sus manos.

4.2. Tarea de apuntado y clic

Al comenzar el ejercicio, se instancia un círculo en una posición pseudo-aleatoria de la pantalla haciendo uso de una semilla de muestreo. De esta manera, el orden en el que aparecen los círculos, es el mismo para todos los participantes. Entonces, si el usuario clicca en la zona interactiva, en un archivo *CSV* de salida se registra el tiempo en el que se ha hecho clic, la posición en el eje X y en el eje Y donde se ha hecho clic, la posición en el eje X y en el eje Y donde se encuentra el centro del círculo y el acierto o el fallo.

A continuación, si el usuario ha clicado en la posición que ocupa el círculo, se reproduce un sonido que indica que la acción acometida ha sido un éxito, el círculo en el que se ha hecho clic desaparece y, si no era el último círculo en ser instanciado, aparece un nuevo círculo en otra posición pseudo-aleatoria. Si por el contrario, el usuario clicca dentro de la zona interactiva en una posición diferente a la que ocupa el círculo, se reproduce un sonido que indica que se ha fallado el clic y el programa seguirá esperando a que el usuario clique en el círculo que le corresponde. Este proceso se repite hasta que el usuario haya clicado sobre 29 círculos. En ese momento, el programa se cierra solo.

4.3. Tarea de escritura

Una vez comenzada la tarea, aparece una palabra aleatoria de un diccionario, el cual contiene las palabras: *hola*, *queso*, *mandril*, *nuevos*, *zapatilla* y *reloj*. En la parte superior central de la pantalla y, justo debajo, aparece la misma palabra con las letras ocultas por rectángulos de color negro. Además, ocupando la parte central inferior de la pantalla aparece un teclado virtual con las letras del teclado *QWERTY*. El objetivo del usuario es, en el caso de utilizar alguna de las dos técnicas de interacción manual, clicar en el teclado virtual y, en el caso de utilizar el teclado, pulsar en este la letra que le corresponda hasta completar la palabra que aparece en la parte superior de la pantalla. De este modo, al clicar dentro del teclado virtual (o al pulsar una tecla del teclado físico), en un archivo csv de salida, se registra el tiempo en el que se ha hecho clic (o se ha pulsado una tecla), las posiciones en el eje X y en el eje Y donde se ha hecho clic, las posiciones en el eje X y en el eje Y donde se encuentra el centro de de la tecla con la letra deseada, la letra de la tecla en la que se ha clicado (o que se ha pulsado), el acierto (o el fallo) y la letra deseada.

A continuación, si el usuario ha clicado en la posición que ocupa la letra que le corresponde en ese momento (o ha pulsado la tecla que contiene la letra que le corresponde en ese momento), suena un sonido que indica que la acción acometida ha sido un éxito, la letra que se acaba de clicar (o pulsar) se muestra en la palabra oculta y, si era la última letra de la palabra y aún quedan palabras en el diccionario, aparece otra palabra nueva en la parte superior de la pantalla y la misma palabra oculta justo debajo de esta. Si por el contrario, el usuario ha clicado dentro del teclado virtual en una posición diferente a la que ocupa la tecla que se debe clicar (o ha pulsado una tecla diferente a la que le corresponde en ese momento), se reproduce un sonido que indica que la acción acometida es errónea y el programa seguirá esperando a que el usuario clique (o pulse) la letra que le corresponde. Este proceso se repite hasta que el usuario haya deletreado correctamente todas las palabras del diccionario. En ese momento, aparece un mensaje indicando que el test ha finalizado y el programa se cierra automáticamente.

Ejemplo de la tarea de escritura:

La figura 23 muestra la tarea de escritura en ejecución. En ella aparece que la palabra que debe deletrear el usuario es *hola*. De igual manera, se observa que el usuario ha conseguido acertar las letras 'h' y 'o', puesto que aparecen descubiertas en la palabra oculta, por lo tanto, en este momento deberá clicar (o pulsar) la tecla que contiene la letra 'l' para avanzar hacia la terminación del programa.

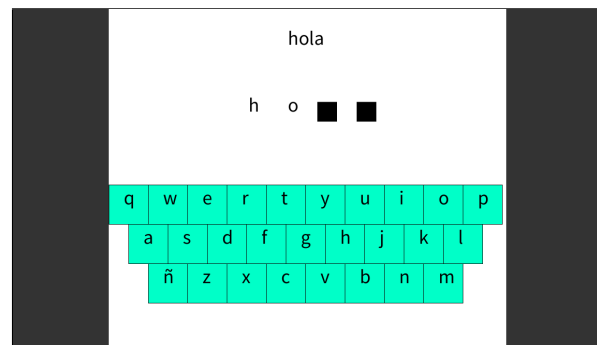


Figura 23: Tarea de escritura en ejecución.

4.4. Procedimiento

Antes de iniciar el experimento, los participantes reciben instrucciones detalladas de las tareas a realizar. En cuanto al orden en el que deben utilizar las diferentes técnicas de apuntado y clic, y escritura para el caso del teclado en la segunda tarea, es determinado antes de iniciar los experimentos por un cuadrado latino, evitando así efectos de orden. De este modo, según se presentan las dos técnicas de entrada de gestos manuales por primera vez, se les explica detalladamente su funcionamiento a los participantes.

Las tareas han sido completadas una después de otra en una única sesión por participante, esto es debido a que la brevedad de ambas tareas no genera demasiada fatiga en los participantes. Por ello, ambas tareas pueden completarse en conjunto en 10 minutos aproximadamente, incluyendo las explicaciones necesarias para llevarlas a cabo.

4.5. Mediciones

Los valores objetivos medidos han sido el tiempo de finalización de la tarea, en inglés *task completion time (TCT)*, y el porcentaje de acierto obtenido por cada una de las tres técnicas utilizadas en los experimentos, también se han cumplimentado cuestionarios subjetivos por los participantes con respecto a la carga de trabajo y la usabilidad, es decir, con respecto al índice de carga de las tareas.

Una vez iniciado el experimento, los participantes se han sentado frente al ordenador con los sistemas *Leap Pointer* y *TouchFree* instalados en él, con la única restricción de utilizar las técnicas correspondientes según su orden en cada momento. Sin embargo, los usuarios han podido apoyar los codos sobre la mesa en la que se ha encontrado el ordenador, utilizar una única mano al usar la técnica de *Leap Pointer* y/o situarse más o menos cerca de la pantalla.

Respecto al cuestionario subjetivo, las preguntas han sido las siguientes: P1-El esfuerzo mental necesario para usar este método es muy alto; P2-El esfuerzo físico necesario para usar este método es muy alto; P3-La dinámica de las actividades con este método ha sido muy rápida; P4-He tenido éxito realizando las actividades con este método; P5-He tenido que invertir mucho esfuerzo para llevar a cabo las actividades con este método; P6-He sentido mucha frustración realizando las actividades con este método; P7-Me gustaría usar este método con frecuencia; P8-El método es innecesariamente complejo; P9-El método era sencillo de usar; P10-Necesitaría la ayuda de un especialista para usar este método; P11-Las diversas funciones del método están muy bien entrelazadas; P12-Había demasiadas inconsistencias en este método; P13-La mayoría de personas podría aprender a usar este método con rapidez; P14-He encontrado el método muy aparatoso; P15-Me he sentido muy seguro usando el sistema; P16-Necesito aprender muchas cosas para empezar a usar este sistema. Los participantes han indicado su grado de acuerdo con las afirmaciones anteriores valorándolas del 1 (totalmente en desacuerdo) al 7 (totalmente de acuerdo).

En la última parte del cuestionario, se les ha pedido a los participantes que puntúen las técnicas de interacción manual, así como el ratón y/o el teclado, en una escala del 0 (técnica pésima) al 10 (técnica excelente).

4.6. Resultados

Las medidas del estudio de usuarios se han analizado mediante *ANOVA* con medidas repetidas para detectar efectos significativos de la condición, y se han usado pruebas *post hoc* con corrección de *Bonferroni* para determinar diferencias significativas.

4.6.1. Resultados Objetivos

Los resultados objetivos muestran información sobre el tiempo promedio de finalización de tareas, en inglés *task completion time (TCT)*, que les ha llevado a los usuarios completar las pruebas con cada técnica. Por otro lado, se ha calculado el porcentaje de acierto medio obtenido por cada una de las tres técnicas en cada tarea.

Como puede observarse en la gráfica de la figura 24, y tal y como cabía esperar, los usuarios han podido completar las tareas mucho más rápido con el ratón y el teclado que con las técnicas de interacción manual, habiendo tardado 19056ms de media para la primera tarea y tan solo 10995ms de media para la segunda. Sin embargo, completar las tareas con la técnica de *Leap Pointer* ha demostrado resultar más veloz que hacerlo con la desarrollada por *Ultraleap*; *TouchFree*. Es más, si bien en la tarea de apuntado y clic, la diferencia de tiempos no es demasiado notoria, habiendo completado la tarea en 44793ms con *Leap Pointer* y en 51745ms con *TouchFree*, sí lo es en la tarea de escritura, donde, se ha realizado la tarea en 41303ms con *Leap Pointer* y en 58796ms con *TouchFree*.

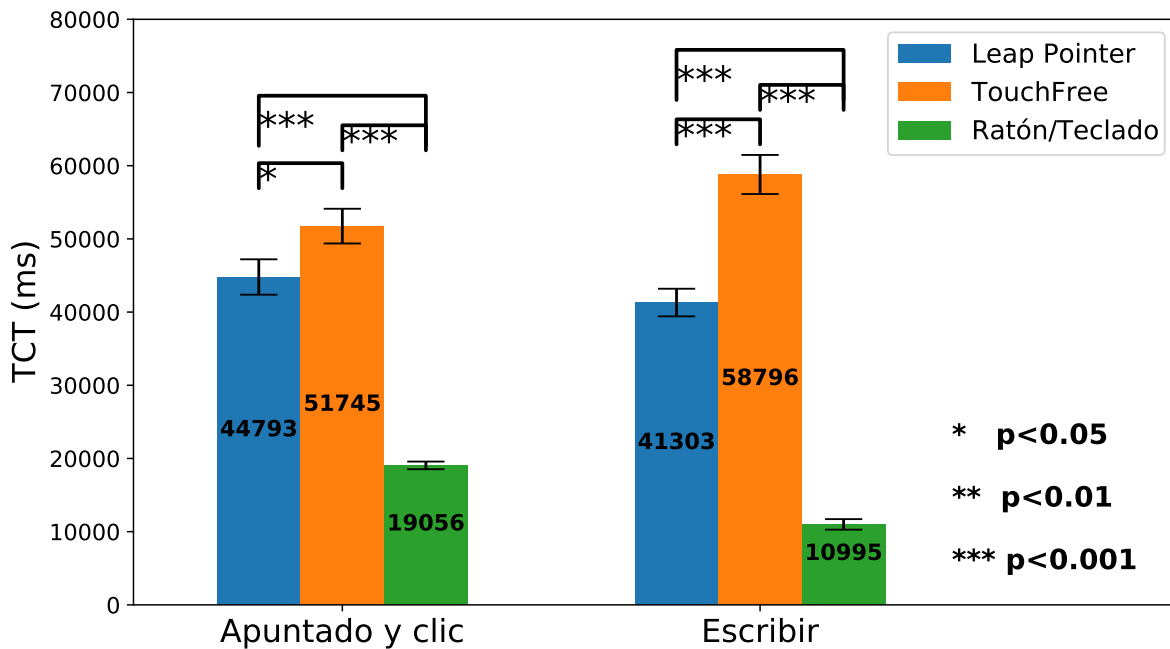


Figura 24: *TCT* medio en milisegundos de cada técnica: *Leap Pointer*, *TouchFree* y Ratón/Teclado; en cada tarea: Apuntado y clic y Escritura.

Respecto a la gráfica de la figura 25, una vez más, los participantes han obtenido mejores resultados utilizando el ratón y el teclado, cometiendo menos errores con estos al lograr un 98.55 % de acierto para la primera tarea y un 95.75 % para la segunda. Con la técnica de *TouchFree*, han pasado de un 87.77 % de acierto en la tarea de apuntado y clic a un 91.77 % en la tarea de escritura, siendo el aumento de porcentajes bastante pequeño en comparación al caso de *Leap Pointer*, donde los resultados han variado de un 76.50 % a un 87.52 % de acierto, lo que supone un aumento en el índice de acierto entre tareas del 14.41 % aproximadamente. Estos aumentos en los índices de acierto, podrían deberse a que las tareas siempre han sido ejecutadas en el mismo orden; primero la de apuntado y clic y posteriormente, la de escritura. Por lo tanto, que el porcentaje de acierto aumente significativamente entre la primera tarea y la segunda, podría denotar que algunos de los usuarios han tardado más que otros en adaptarse a las diferentes técnicas de interacción manual satisfactoriamente, reflejando, por ende, curvas de aprendizaje.

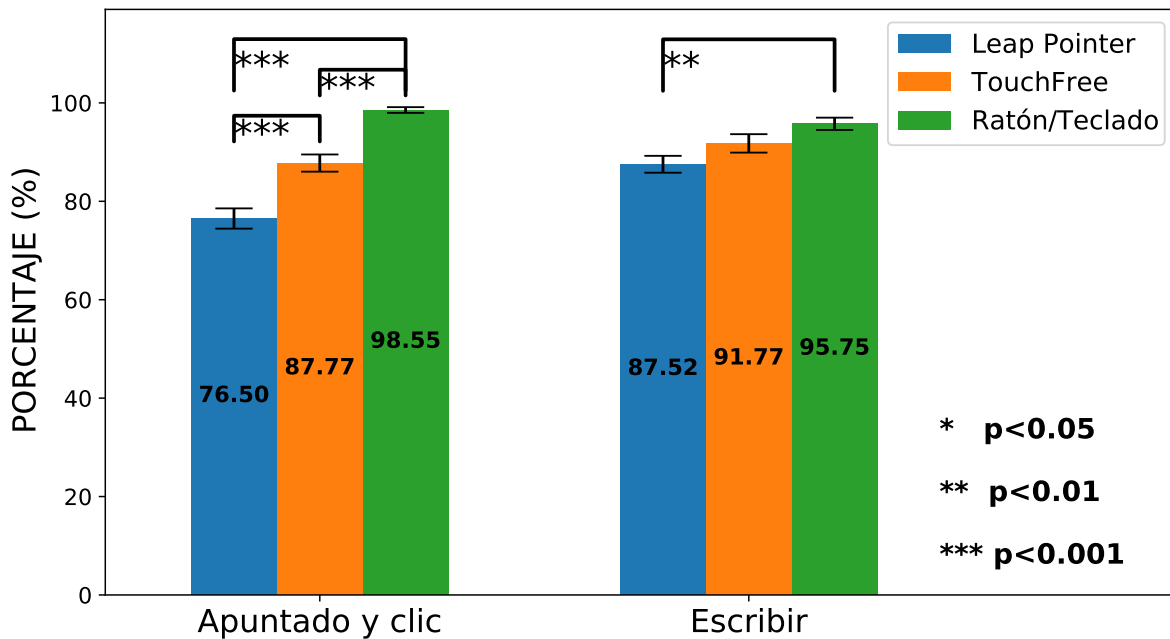


Figura 25: Porcentaje medio de acierto de cada técnica: *Leap Pointer*, *TouchFree* y Ratón/Teclado; en cada tarea: Apuntado y clic y Escritura.

En las gráficas que se muestran a continuación, aparecen expuestos los resultados de ambas tareas con las técnicas de apuntado y clic: *Leap Pointer*, *TouchFree* y el ratón, nótese que el teclado se ha obviado ya que no pertenece al subconjunto de técnicas de apuntado y clic. Asimismo, en estas gráficas se ha marcado con una x de color verde los clics acertados en cada momento, y con una x de color rojo, los clics fallados. Además las líneas discontinuas muestran la distancia entre la posición donde se ha clicado y la posición del punto objetivo de cada momento.

En el caso de la primera tarea, además, se ha indicado numéricamente el orden en el que han ido apareciendo los círculos (puntos objetivo) y, en el caso de la segunda tarea, como añadido, se muestra la letra objetivo en la posición de destino, es decir, en la posición que ocupa el punto objetivo.

Como nota al margen, de las gráficas resultantes de la primera tarea, se observa que *Leap Pointer* ha fallado más al tener que clicar en las esquinas. Esto es debido a que faltaría calibrar esta técnica, algo que se ha contemplado en las líneas futuras de este trabajo.

4.6.2. Visualización de los resultados de la tarea de apuntado y clic

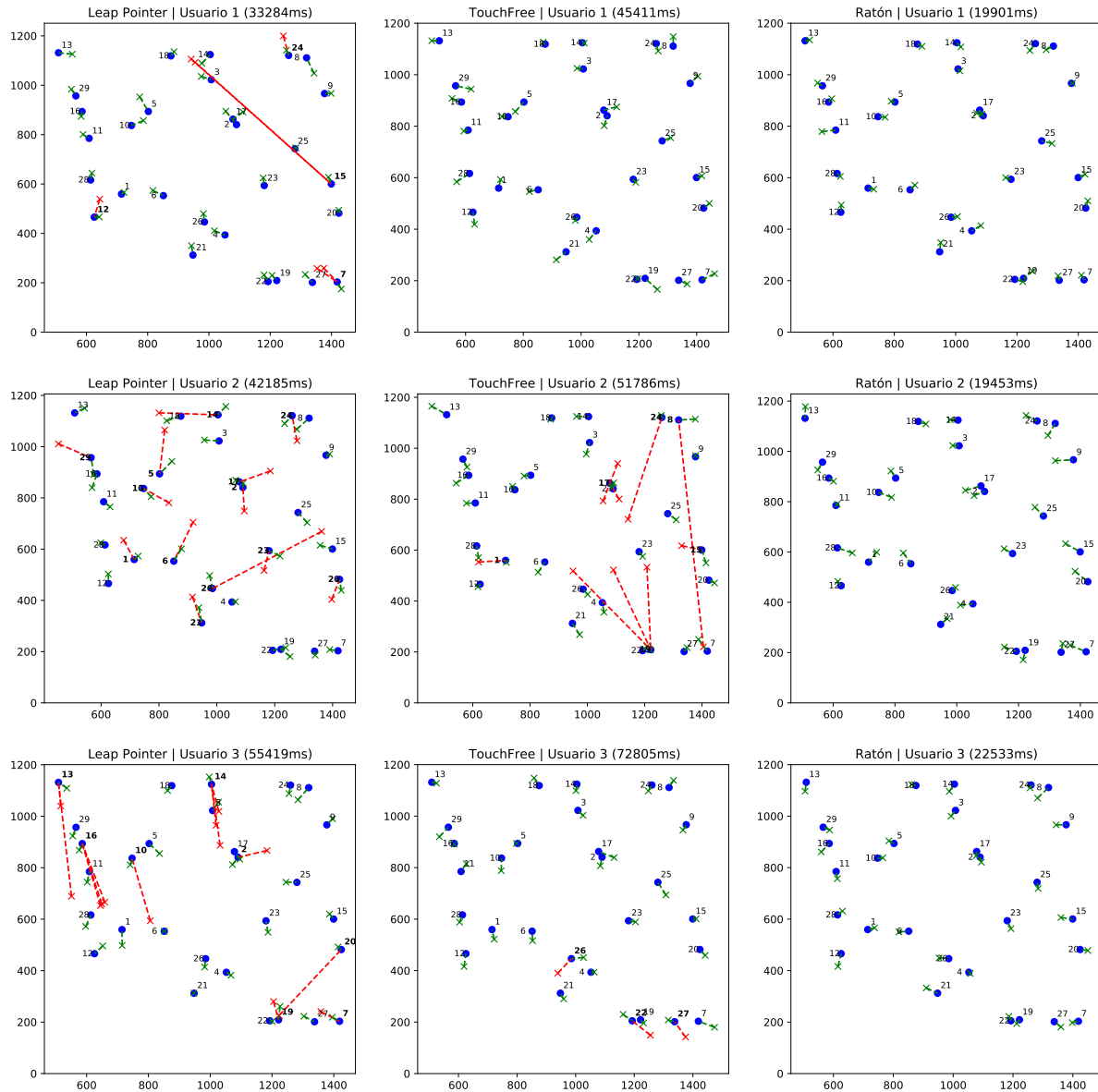


Figura 26: Resultados de los usuarios, del uno al tres, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: *Leap Pointer*, *Ultraleap's TouchFree* y *Ratón*.

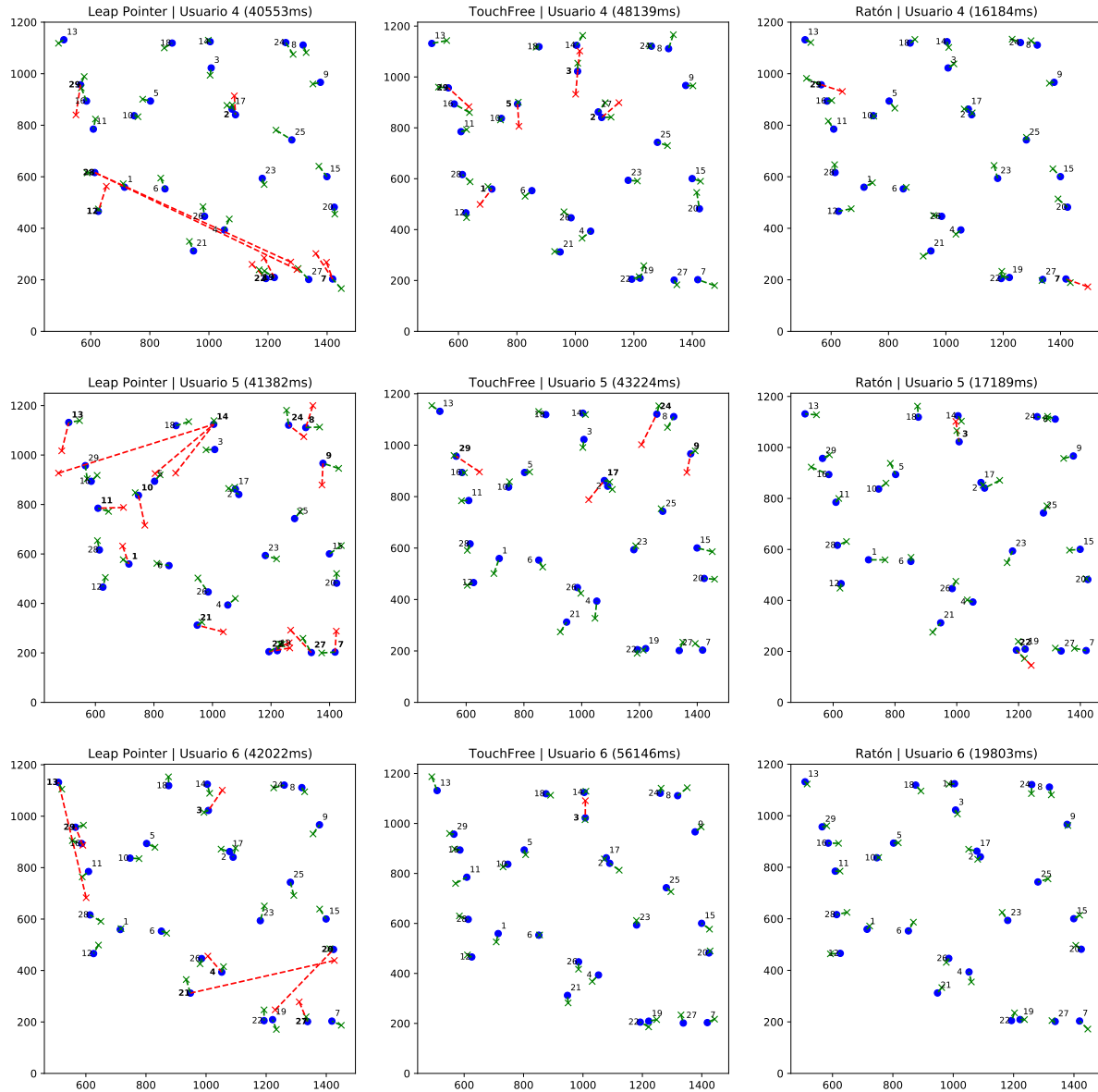


Figura 27: Resultados de los usuarios, del cuatro al seis, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: *Leap Pointer*, *Ultraleap's TouchFree* y *Ratón*.

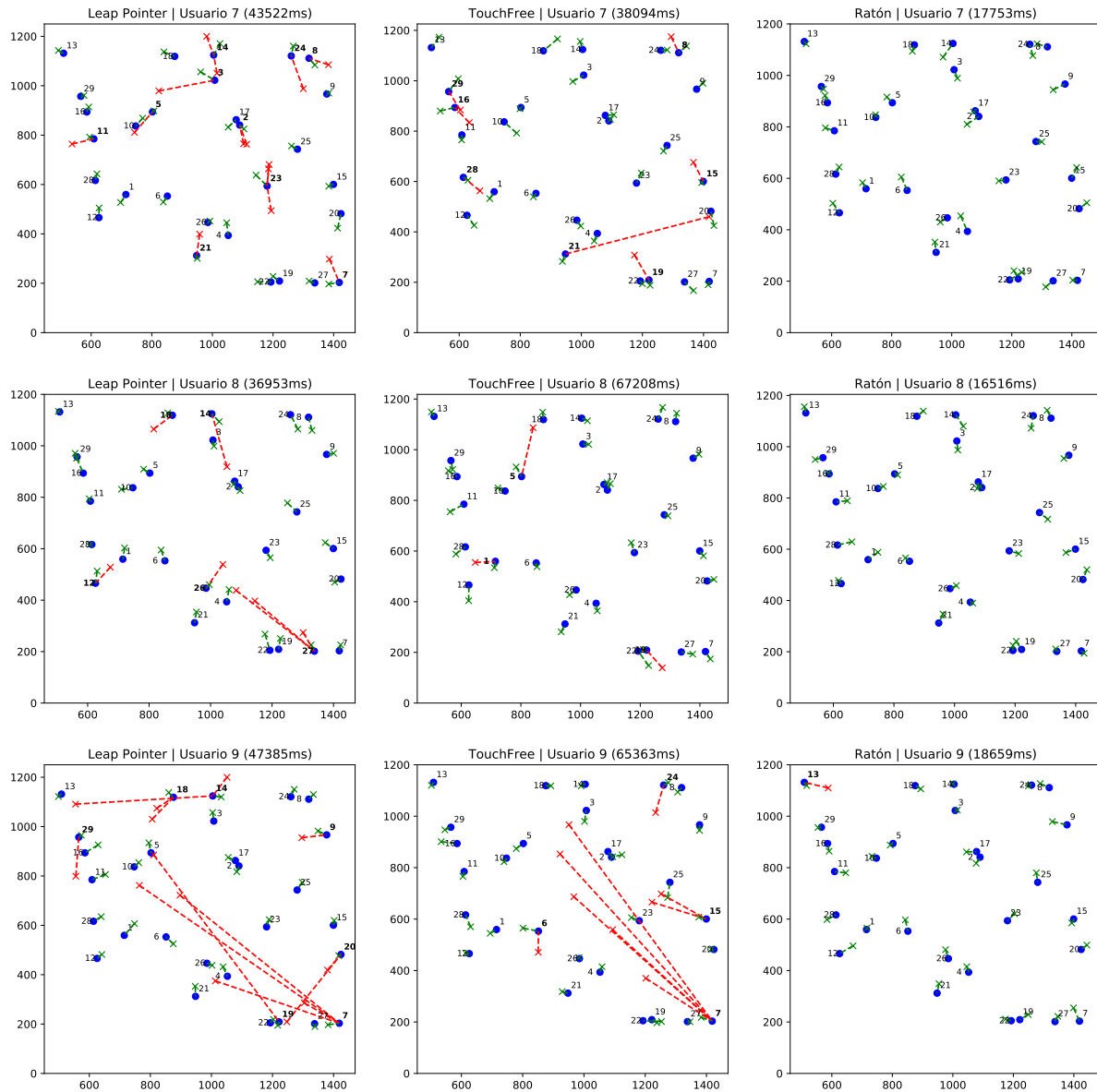


Figura 28: Resultados de los usuarios, del siete al nueve, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: *Leap Pointer*, *Ultraleap's TouchFree* y *Ratón*.

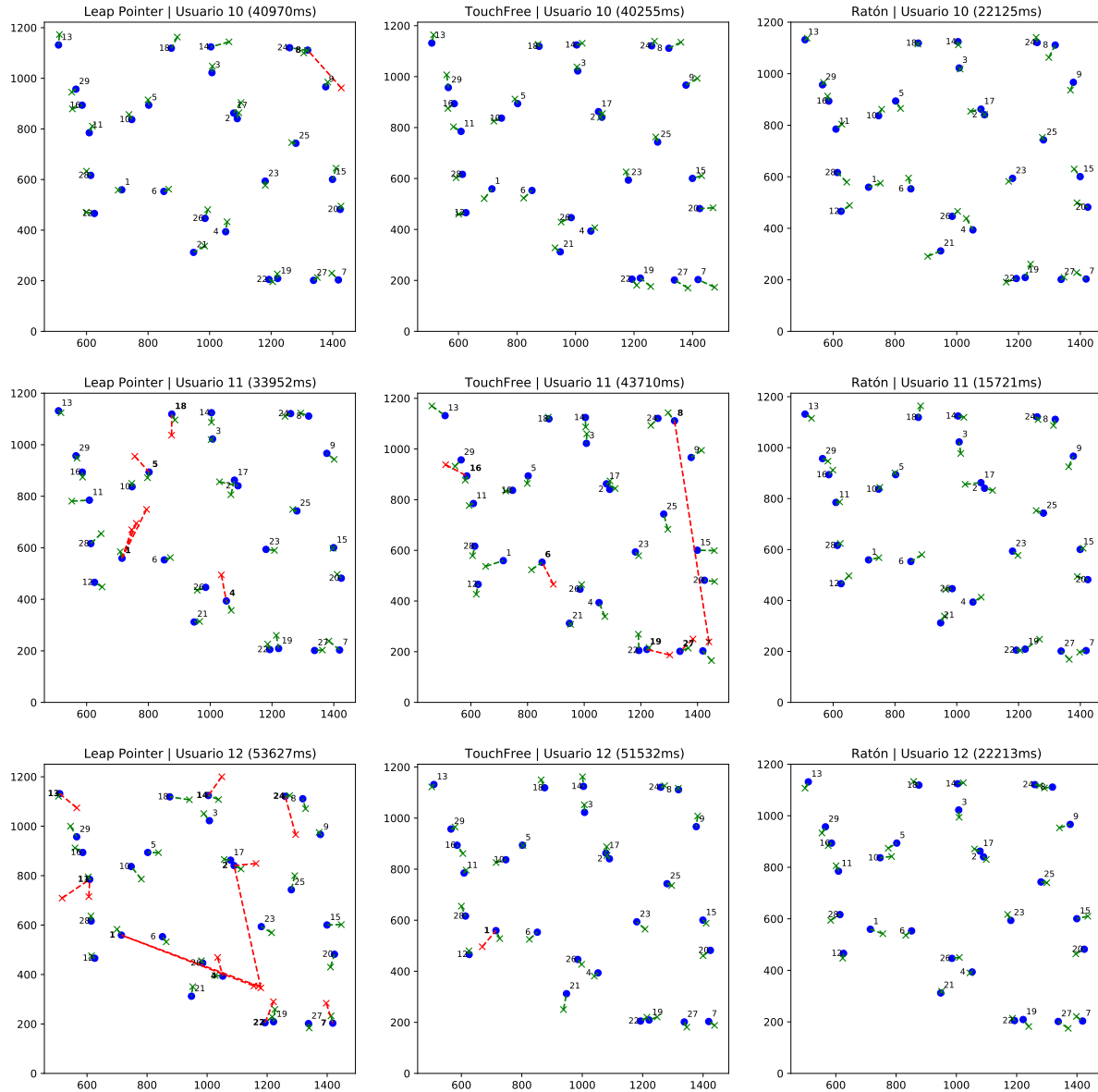


Figura 29: Resultados de los usuarios, del diez al doce, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: *Leap Pointer*, *Ultraleap's TouchFree* y *Ratón*.

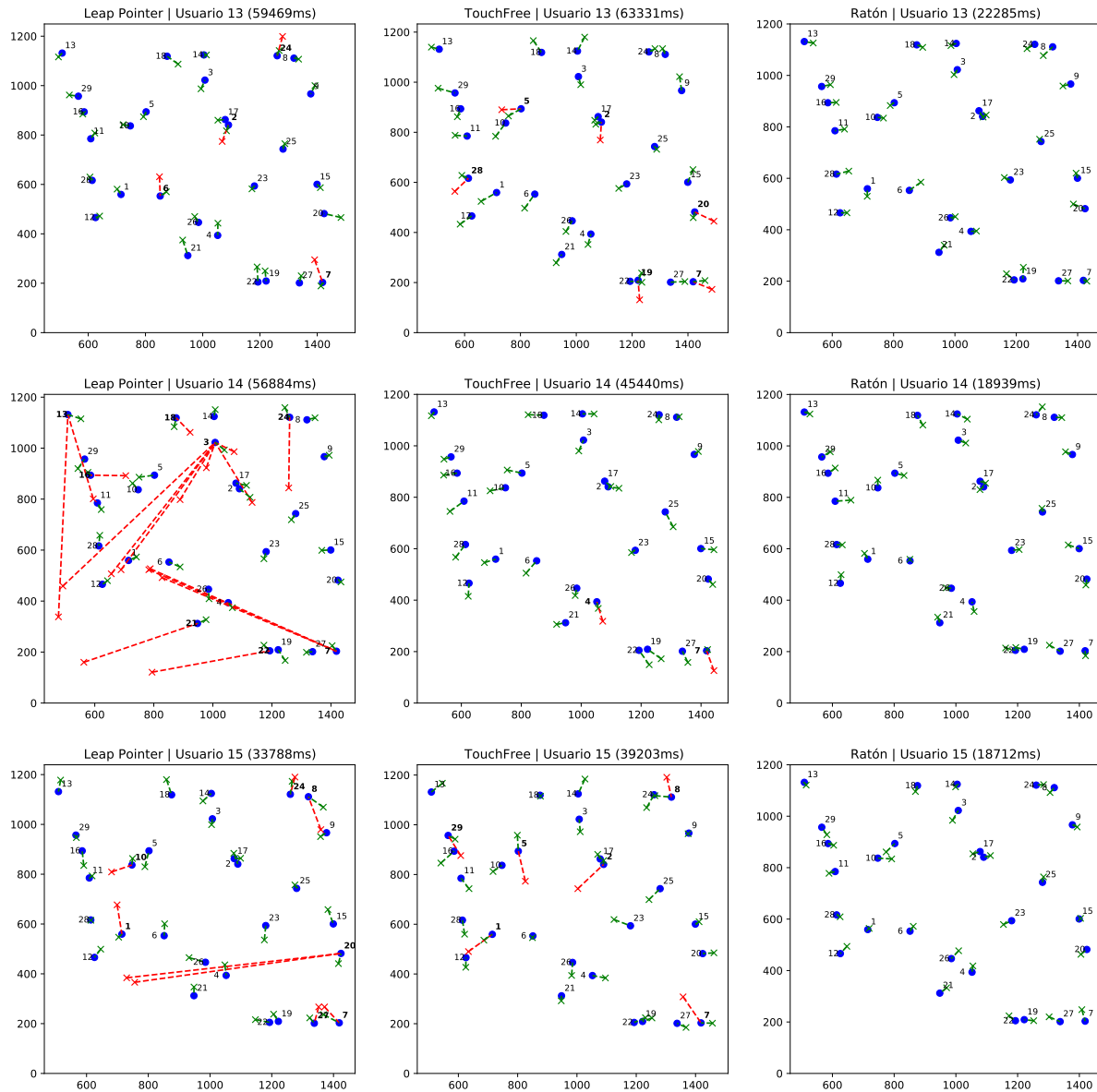


Figura 30: Resultados de los usuarios, del trece al quince, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: *Leap Pointer*, *Ultraleap's TouchFree* y *Ratón*.

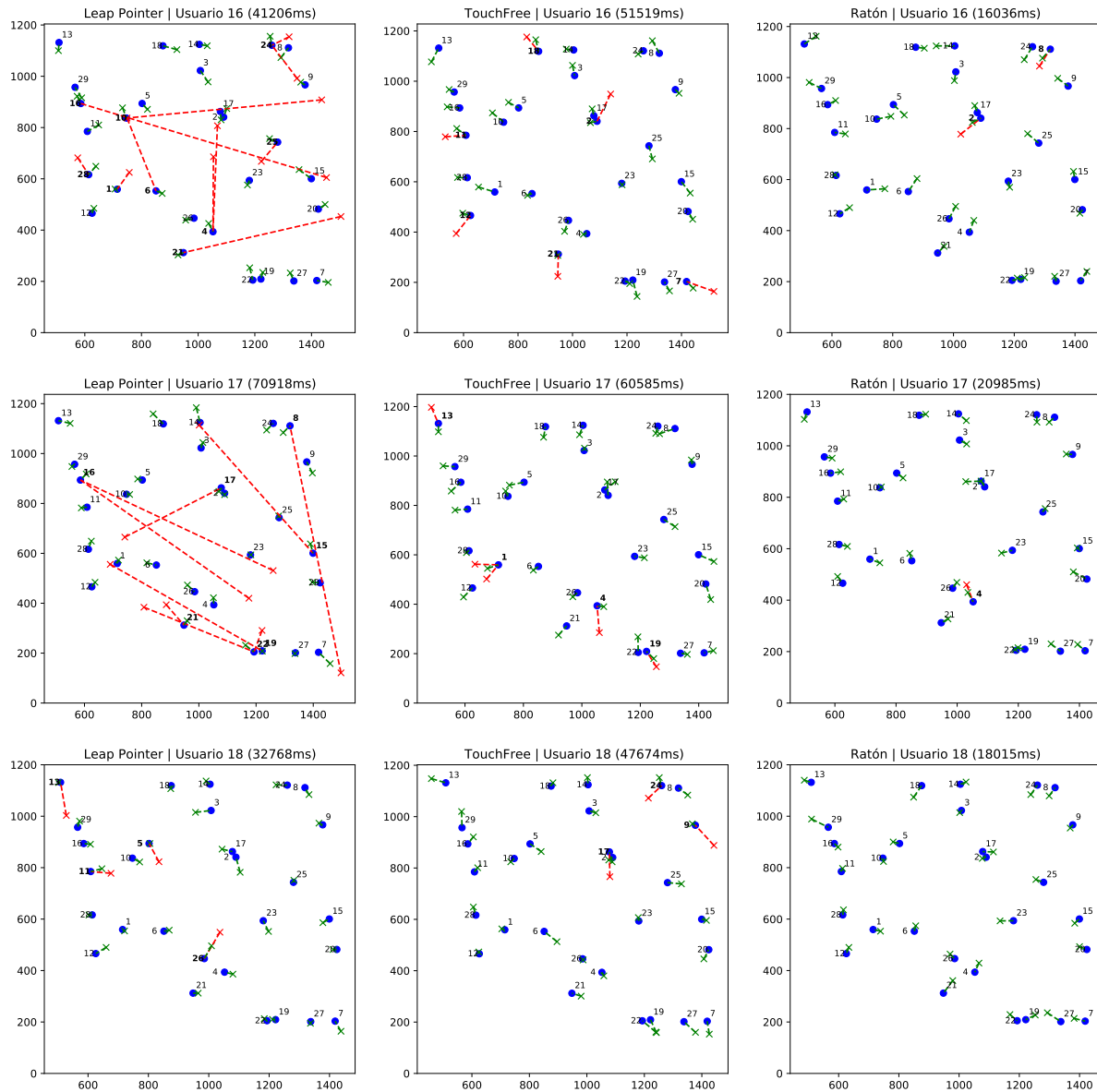


Figura 31: Resultados de los usuarios, del dieciséis al dieciocho, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: *Leap Pointer*, *Ultraleap's TouchFree* y *Ratón*.

4.6.3. Visualización de los resultados de la tarea de escritura

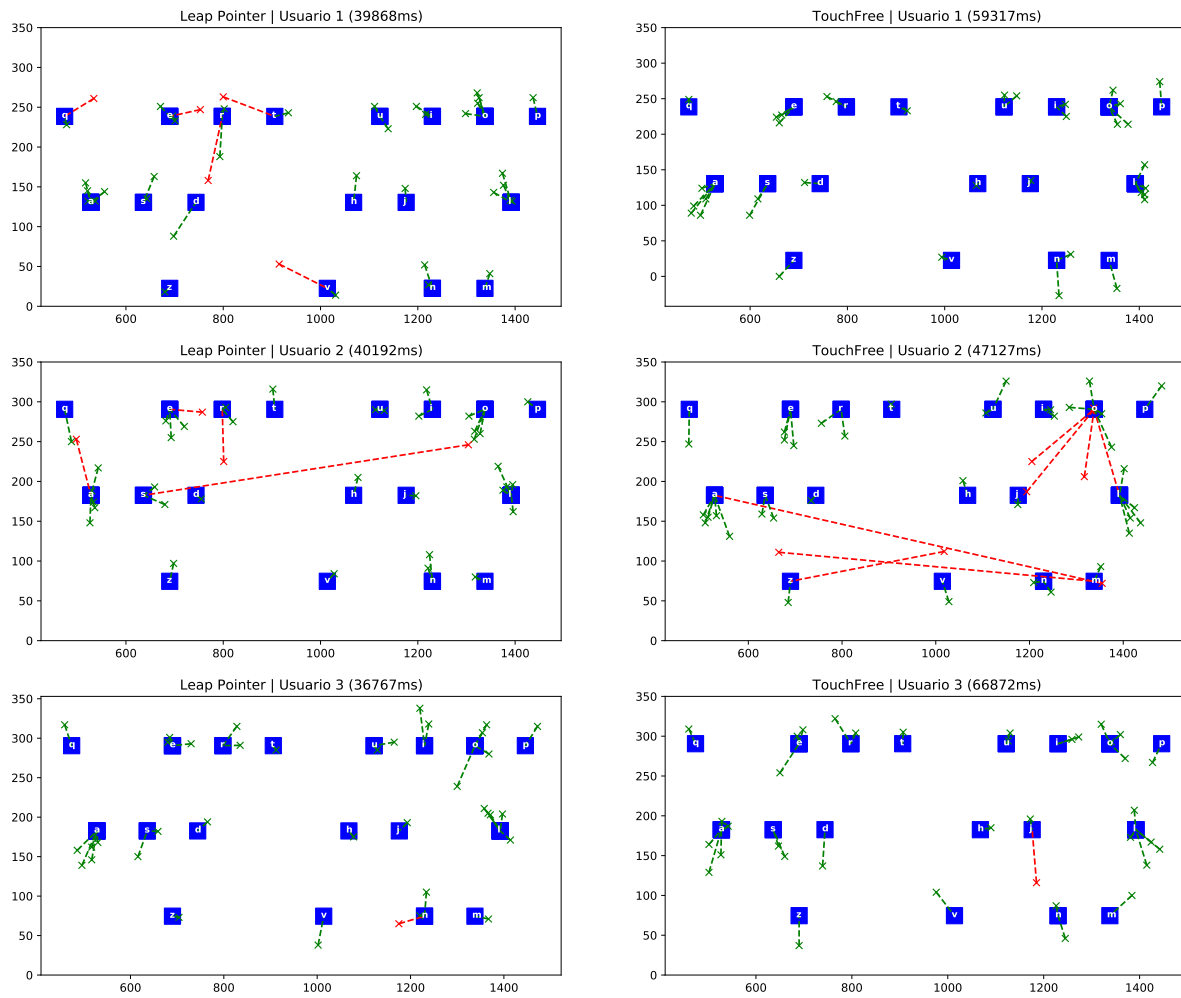


Figura 32: Resultados de los usuarios, del uno al tres, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: *Leap Pointer* y *Ultraleap's TouchFree*.

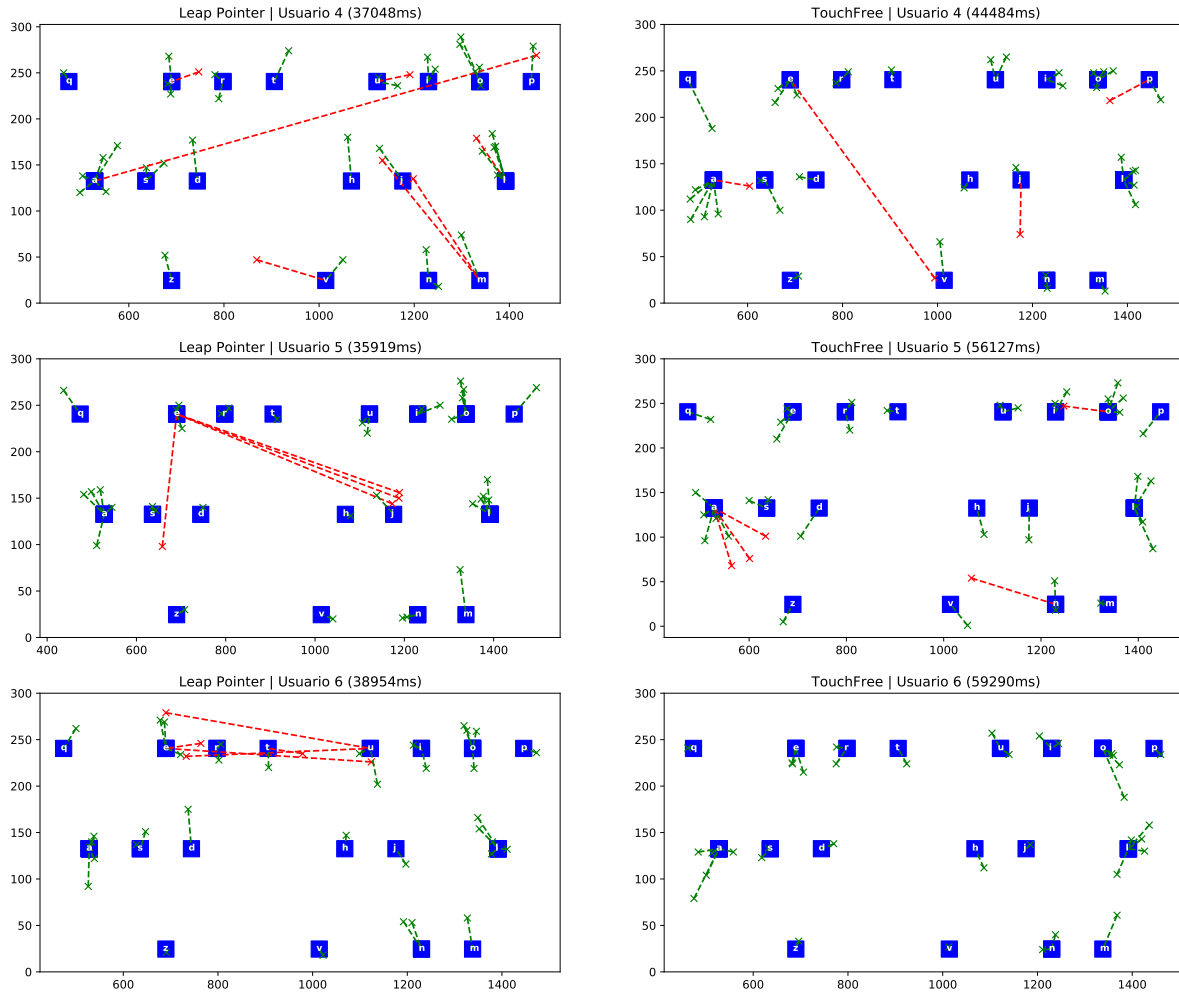


Figura 33: Resultados de los usuarios, del cuatro al seis, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: *Leap Pointer* y *Ultraleap's TouchFree*.

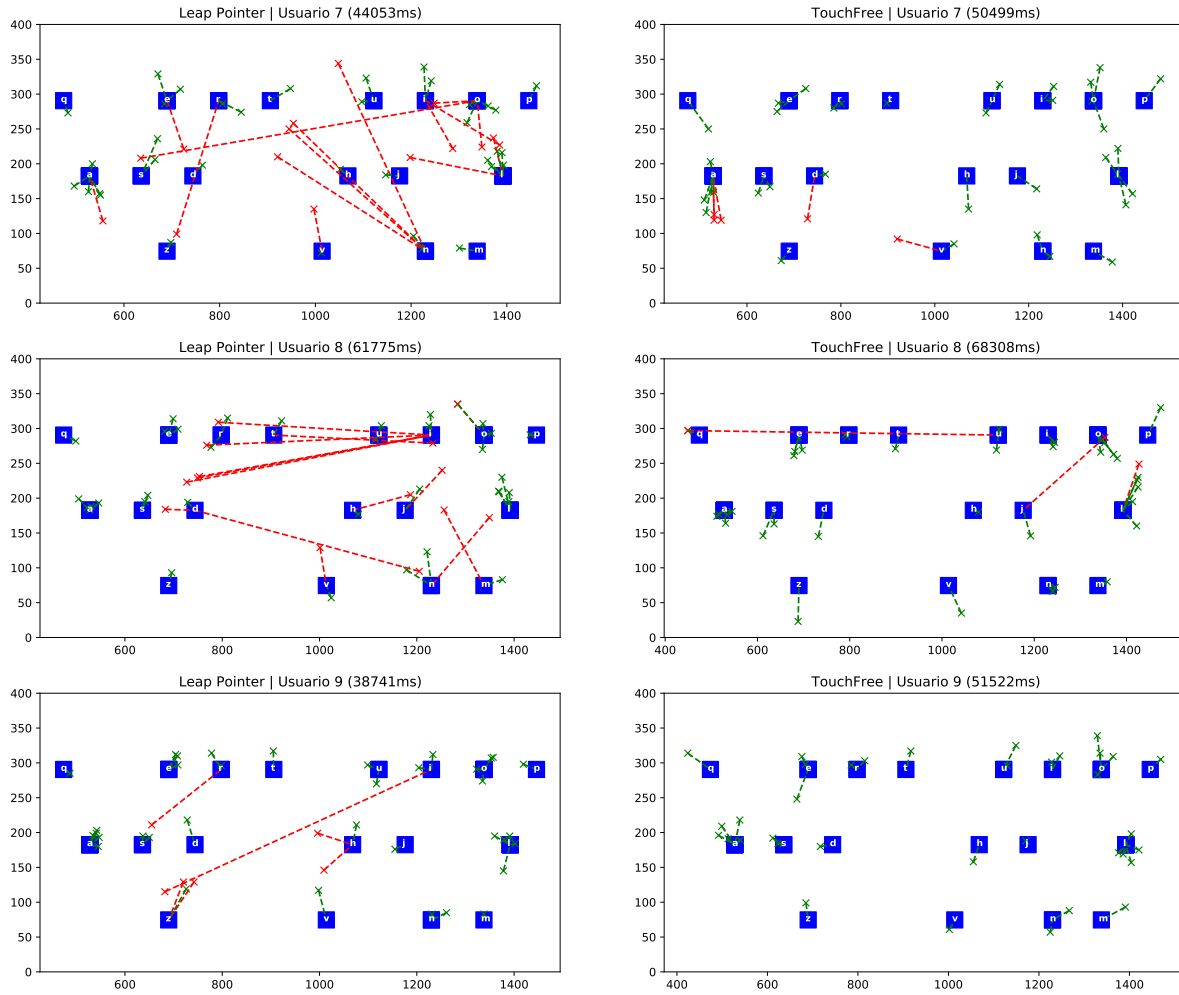


Figura 34: Resultados de los usuarios, del siete al nueve, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: *Leap Pointer* y *Ultraleap's TouchFree*.

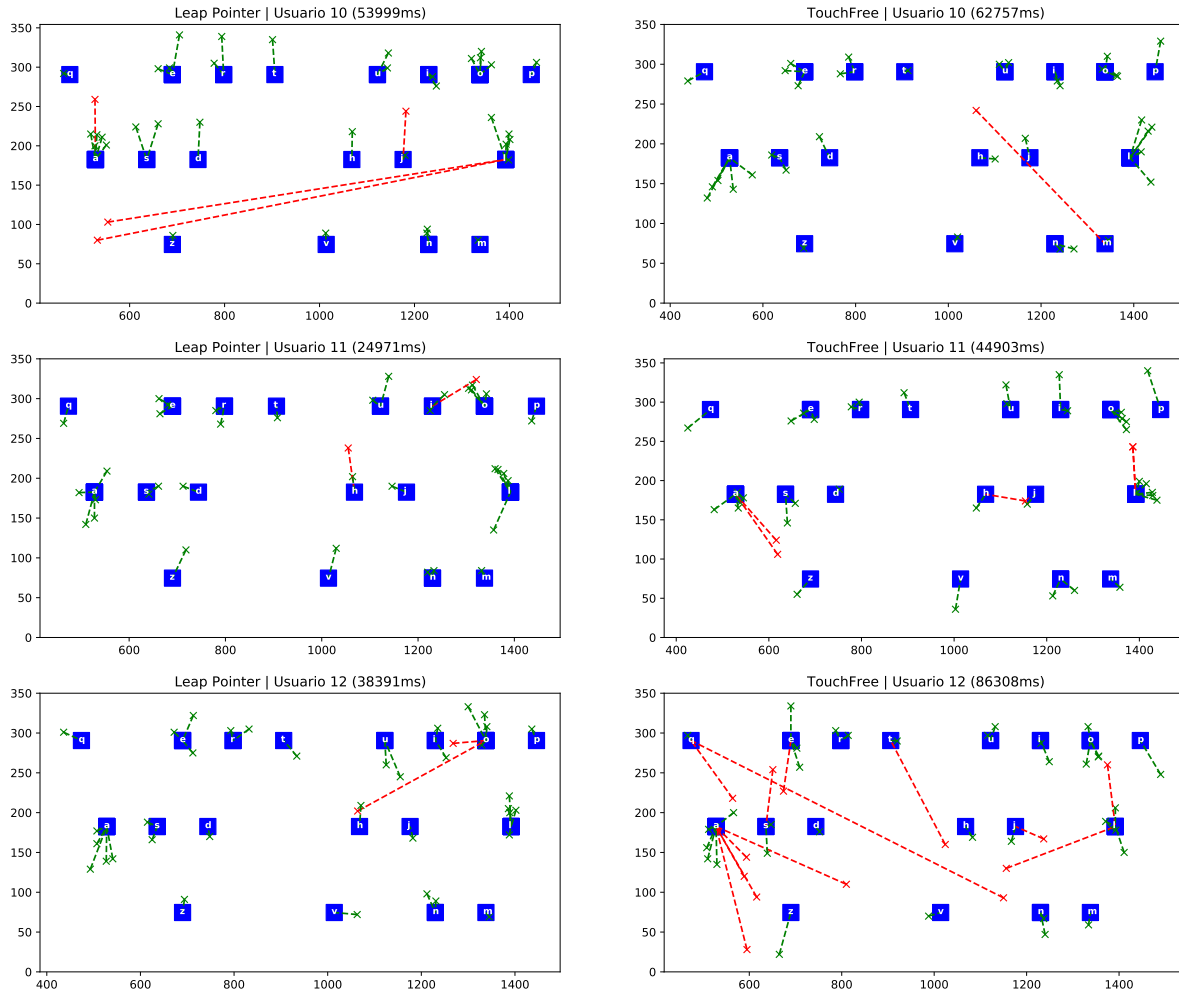


Figura 35: Resultados de los usuarios, del diez al doce, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: *Leap Pointer* y *Ultraleap's TouchFree*.

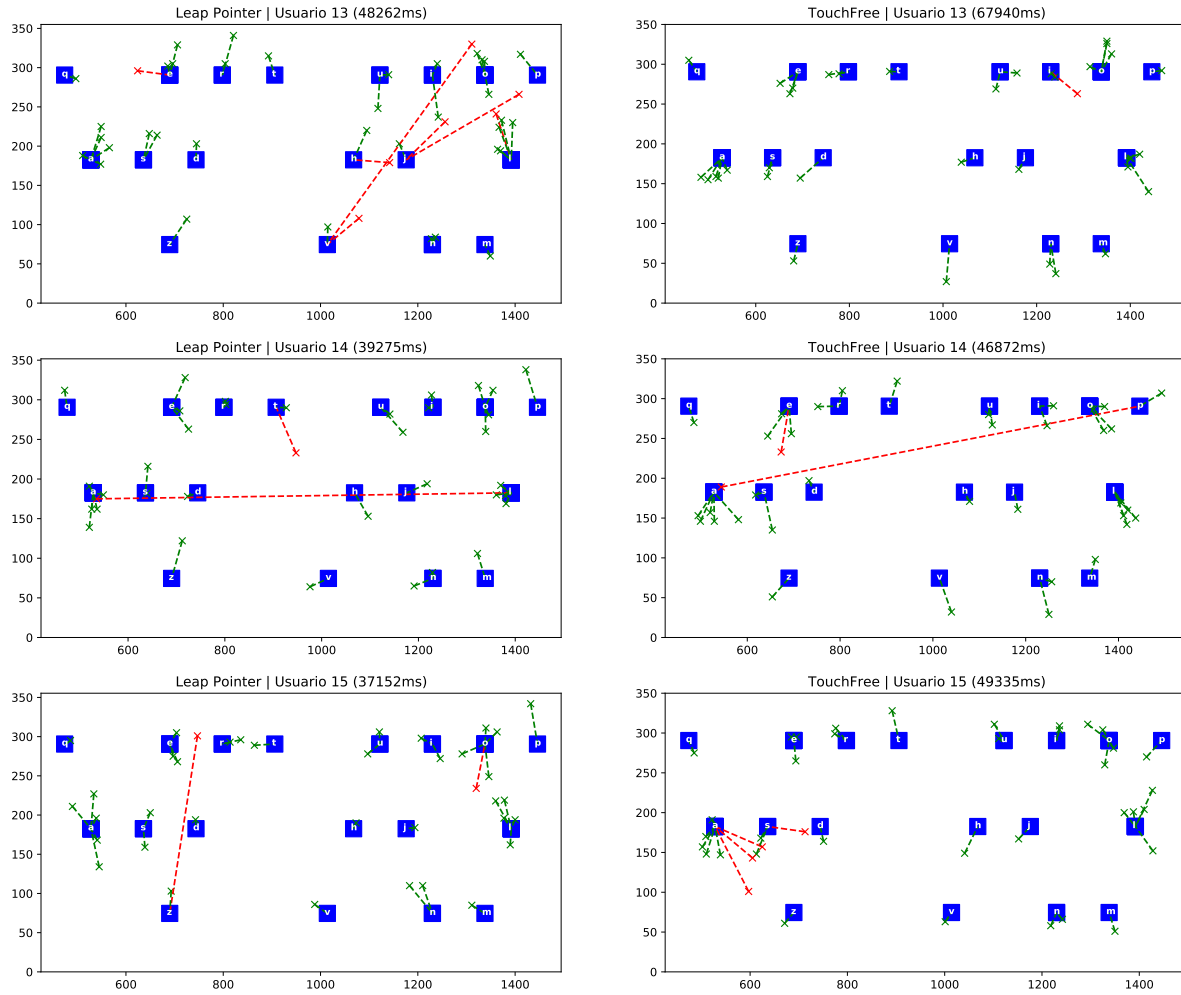


Figura 36: Resultados de los usuarios, del trece al quince, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: *Leap Pointer* y *Ultraleap's TouchFree*.

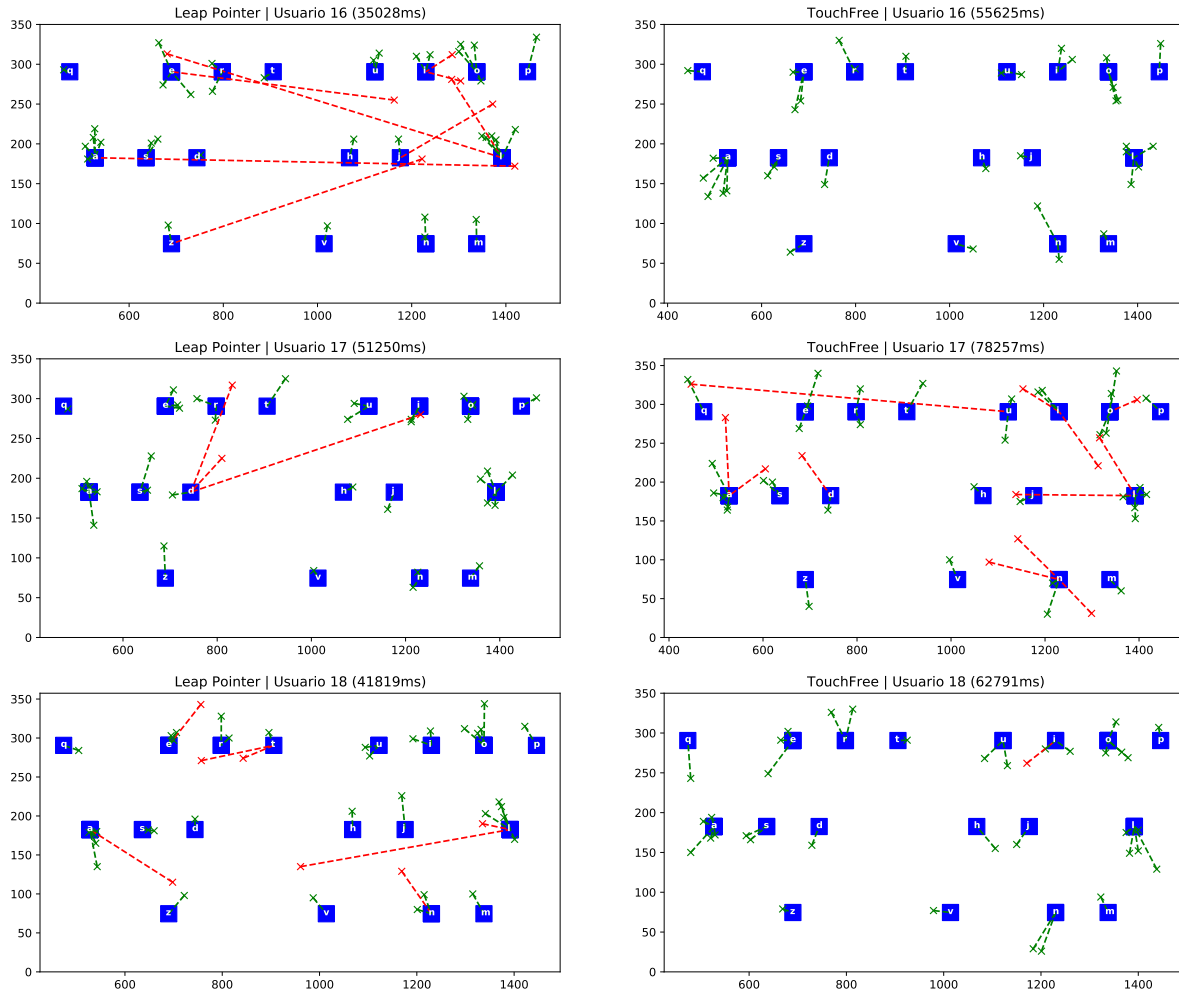


Figura 37: Resultados de los usuarios, del dieciséis al dieciocho, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: *Leap Pointer* y *Ultraleap's TouchFree*.

4.6.4. Resultados Subjetivos

A continuación se muestran la puntuación media junto con sus varianzas obtenidas en cada pregunta de la parte subjetiva.

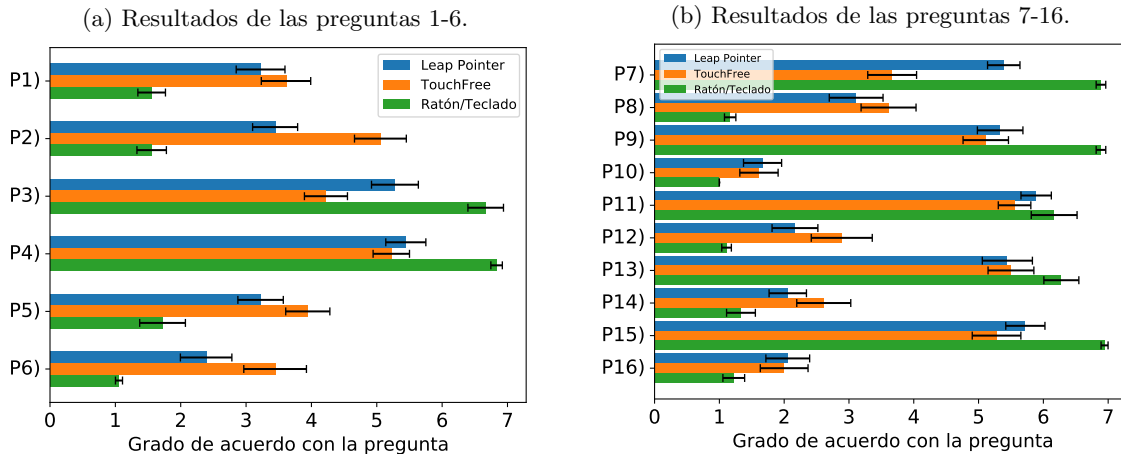


Figura 38: Puntuación media de cada pregunta del cuestionario subjetivo.

4.6.5. Interpretación

Respecto al esfuerzo mental necesario para utilizar cada método, el ratón y el teclado han resultado ser las mejores opciones para evitar el agotamiento mental, los usuarios no han tenido que preocuparse en pensar como funcionan, o mejor dicho, cómo se utilizan estos métodos, ya que están habituados a trabajar con estos. A esta técnica le precede la de *Leap Pointer* y finalmente, la de *TouchFree* resulta ser la más agotadora mentalmente.

Este mismo orden se manifiesta en lo que ha cansancio físico se refiere, siendo el ratón la opción menos agotadora, precedido por *Leap Pointer* y seguido por *TouchFree* a la cola. Cabe destacar, que varios de los participantes han justificado el cansancio acumulado por *TouchFree* debido a que, para clicar, hay que trasladar todo el brazo desde atrás hacia delante y desde adelante hacia atrás para realizar un simple clic, por lo tanto, tener que repetir este movimiento una y otra vez ha resultado ser agotador.

Es notorio, además, que los participantes han sido perfectamente conscientes del tiempo que les ha llevado ejecutar las tareas con cada técnica, pues han supuesto que la dinámica de las actividades: con el método del teclado y el ratón ha sido muy rápida; con *Leap Pointer* ha sido bastante rápida; con *TouchFree* ha sido más lenta en comparación al resto. Lo cual, concuerda con el *TCT* de cada técnica.

Por el contrario, los participantes han creído lograr mayor éxito realizando las actividades con *Leap Pointer* que con *TouchFree*, lo cual se contradice con el porcentaje de acierto obtenido con estas técnicas, siendo mayor para el caso de *TouchFree*, luego, podría decirse que los participantes han sido optimistas ha la hora de valorar los resultados obtenidos con *Leap Pointer*.

En cuanto a la frustración que les ha supuesto a los usuarios utilizar cada método, *TouchFree* se lleva la palma, debido a que: La lentitud con la que se han realizado las tareas en contraste con el resto de técnicas resulta irritante; el cansancio físico que supone usar el método es notorio sin importar el orden en el que se utilicen las diferentes técnicas; es difícil saber donde se encuentra el umbral interactivo para realizar clics; una vez atravesado el umbral, es muy difícil dar marcha atrás la mano sin acometer un clic, siendo este el motivo principal de los clics fallados originados por esta técnica.

En lo que ha usabilidad se refiere, si bien ambos métodos de interacción manual pueden parecer más o menos complejos en un principio, dependiendo de la persona, resultan sencillos de utilizar.

Con todo esto, los participantes han valorado las diferentes técnicas de una forma más genérica puntuándolas del 0 (técnica pésima) al 10 (técnica excelente). La media de las puntuaciones es la observada en la figura 39.

En definitiva, *Leap Pointer* ha sido la técnica de interacción manual favorita entre los participantes.

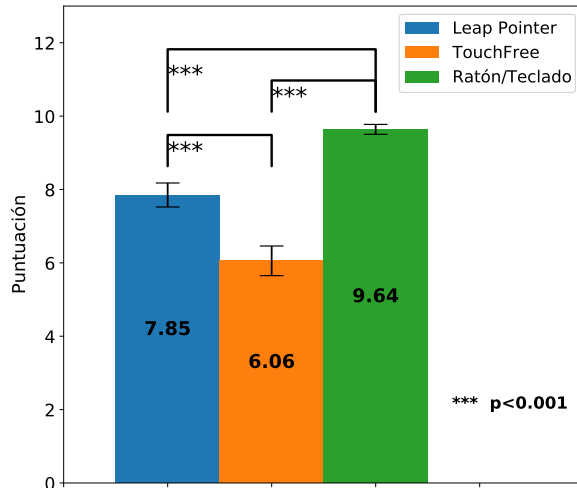


Figura 39: Puntuaciones medias de cada técnica.

4.7. *Discusión*

El ratón y el teclado presentan una solución casi perfecta para comunicarse con el ordenador a modo de *input*. Sin embargo, como bien recalca la premisa de este trabajo, cuando se trata de compartir un mismo ratón y/o teclado con un número indefinido de personas, la solución no es higiénica, y por tanto, no es una solución saludable.

Analizando entonces las técnicas de interacción manual, gracias a los resultados reflejados anteriormente, es destacable que los usuarios de dichas técnicas son perfectamente amoldables a su utilización con algo de práctica. Es más, si bien los resultados de precisión siguen siendo mejorables, la técnica de *Leap Pointer* ha demostrado ser una técnica digna competidora del ratón, aunque no del teclado.

Por último, en lo que a la precisión del método desarrollado respecta, habría que perfeccionarla con precaución, ya que *TouchFree* parece sacrificar tiempo y velocidad por precisión y, sin embargo, es una de las razones de peso por la cual se genera frustración entre sus usuarios. Es más, este factor sumado al hecho de no ser bimanual y delegar todo el esfuerzo físico sobre un único brazo, acaban otorgando otro punto a favor de la técnica *Leap Pointer*.

5. Conclusiones

El estudio de usuario ha demostrado que el sistema *Leap Pointer* presenta una gran potencia para manifestarse como una técnica de interacción manual, la cual es perfectamente usable y no deja demasiada fatiga en los brazos a la hora de realizar tareas de apuntado y clic. Además, gracias a todas las funcionalidades de las que se compone el sistema, este programa aporta beneficios tanto a particulares como a instituciones públicas.

Por un lado, en lo que al uso de un particular respecta, el sistema permite trabajar de forma híbrida junto con el ratón, gracias a la función de activación y desactivación. De esta manera, se plantea la prevención de uno de los riesgos laborales más comunes entre informáticos: la prevención de padecer el síndrome del túnel carpiano. El uso bimanual de *Leap Pointer* es el que responde a esta prevención laboral, es decir, al poder alternar el gesto para clicar entre ambas manos, se proporciona un espacio para el descanso y la recuperación del cansancio de los trabajadores. Asimismo, esta técnica habilita posturas más orgánicas, concediendo la opción de trabajar de pie, evitando así dolores de espalda y disminuyendo el nivel de fatiga de los usuarios. Por ello, estos puntos favorecen al número de horas productivas de los usuarios a la hora de trabajar con el *PC*.

Por otro lado, en lo que al uso dentro del marco de las instituciones públicas respecta, esta técnica presenta una respuesta tan útil como higiénica para abordar el problema de la premisa original. En casos donde varios individuos tengan que utilizar la misma pantalla táctil, por ejemplo, al consultar información en un museo o a la hora de tener que sacar dinero de un cajero; *Leap Pointer* propone una alternativa perfectamente válida para evitar el contacto directo de estas pantallas, mejorando de esta manera la higiene de cada persona; y, por ende, mejorando la salud pública.

Finalmente, cabe la posibilidad de usar el sistema *Leap Pointer* junto con tareas interactivas, como las propias tareas utilizadas en el estudio de usuario de este trabajo, a modo de una herramienta eficaz para trabajar la motricidad fina de aquellas personas que se encuentren en algún proceso de rehabilitación o, que sencillamente, busquen mejorar su coordinación motora en movimientos pequeños y precisos mediante el gesto de pinza.

6. Líneas Futuras

Con el fin de mejorar el *feedback* hacia el usuario, sería interesante fabricar dispositivos hápticos que simulasen los punteros del programa. De esta manera el usuario podría medir mejor la distancia con la que puede, o no, hacer clic.

Así pues, para una fase inicial bastaría con utilizar un material elástico esférico que además fuera capaz de no interferir en la captura de imágenes de la *Leap Motion*.

Posteriormente, yendo más allá, se podría incluso utilizar un material inteligente que tuviese una dureza variable. Este efecto podría lograrse replicando las diferentes condiciones de las bolsas de café: cuando no están sometidas al vacío las bolsas de café son blandas. Sin embargo, al aplicarle vacío, el material se vuelve duro. Así pues, la dureza de este material estaría directamente relacionada con la tarea que se quisiera desarrollar, por ejemplo, querer eliminar una carpeta del ordenador que ocupase una cantidad de gigas considerables podría hacer que el material se volviese duro, y para clicar podría seguir teniendo una dureza suave.

Otro factor mejorable del sistema es la precisión de los punteros. Es más, esta cuestión ya se ha planteado en el estudio de usuario, y recordando lo visto en dicho punto, cabe destacar que existe un dilema con el que hay que lidiar para mejorar la precisión, pues, es difícil mejorarla sin alterar la velocidad de respuesta del programa. Obviando esto, una posible solución, podría ser el hecho de mantener el puntero en una posición fija si los dedos, índice y pulgar, tratan de realizar el gesto de pinza y se encuentran a una distancia mayor que el umbral de clic, pero menor que en la posición de apuntado.

En lo que a las funcionalidades del sistema respecta, podría desarrollarse un método de reconocimiento de gestos para desplegar un teclado virtual, similar al de la tarea de escritura del estudio de usuario; y así habilitar la entrada de texto, ya que se ha demostrado que *Leap Pointer* puede utilizarse para realizar tareas del tipo *text entry*.

Finalmente, podría añadirse un sistema de calibración y otro de ajustes para situar la cámara en diferentes puntos y ángulos no contemplados en este trabajo.

Índice de figuras

1.	Captura de pantalla tomada del vídeo mostrado en Twitter por Jamin Hu [4].	5
2.	Ejecución del programa <i>Leap Pointer</i> en su fase beta.	6
3.	Ejemplo del funcionamiento de <i>Leap Motion</i> [14].	7
4.	<i>Leap Motion</i> deconstruido[18].	8
5.	Diagrama de casos de uso del sistema <i>Leap Pointer</i>	9
6.	Desplazamiento de la mano derecha.	10
7.	Mano derecha realizando un clic.	11
8.	Proceso para desactivar las funcionalidades de la mano derecha.	12
9.	Proceso para activar las funcionalidades de la mano derecha.	13
10.	Modos para posicionar la <i>Leap Motion</i>	14
11.	Representación de cómo se coloca la <i>Leap Motion</i>	15
12.	<i>Prefabs</i> de manos para <i>Unity</i>	15
13.	Composición de la escena en <i>Unity</i>	16
14.	Diagrama de clases del sistema <i>Leap Pointer</i> simplificado.	16
15.	Diagrama de clases de la palma izquierda.	17
16.	Diagrama de clases de la palma derecha.	18
17.	Diagrama de clases del sistema <i>Leap Pointer</i> completo.	19
18.	Desglose del <i>Game Object GhostHands</i> hasta sus palmas.	20
19.	Panel de ajustes de <i>TouchFree</i> para indicar la posición que toma la cámara <i>Leap Motion</i> [22].	21
20.	Panel de ajustes interactivos de <i>TouchFree</i>	22
21.	Desplazamiento de la mano derecha.	23
22.	Mano derecha realizando un clic.	24
23.	Tarea de escritura en ejecución.	26
24.	<i>TCT</i> medio en milisegundos de cada técnica: <i>Leap Pointer</i> , <i>TouchFree</i> y Ratón/Teclado; en cada tarea: Apuntado y clic y Escritura.	28
25.	Porcentaje medio de acierto de cada técnica: <i>Leap Pointer</i> , <i>TouchFree</i> y Ratón/Teclado; en cada tarea: Apuntado y clic y Escritura.	29
26.	Resultados de los usuarios, del uno al tres, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> , <i>Ultraleap's TouchFree</i> y Ratón.	30
27.	Resultados de los usuarios, del cuatro al seis, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> , <i>Ultraleap's TouchFree</i> y Ratón.	31

28.	Resultados de los usuarios, del siete al nueve, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> , <i>Ultraleap's TouchFree</i> y Ratón.	32
29.	Resultados de los usuarios, del diez al doce, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> , <i>Ultraleap's TouchFree</i> y Ratón.	33
30.	Resultados de los usuarios, del trece al quince, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> , <i>Ultraleap's TouchFree</i> y Ratón.	34
31.	Resultados de los usuarios, del dieciséis al dieciocho, en la primera tarea utilizando las diferentes técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> , <i>Ultraleap's TouchFree</i> y Ratón.	35
32.	Resultados de los usuarios, del uno al tres, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> y <i>Ultraleap's TouchFree</i>	36
33.	Resultados de los usuarios, del cuatro al seis, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> y <i>Ultraleap's TouchFree</i>	37
34.	Resultados de los usuarios, del siete al nueve, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> y <i>Ultraleap's TouchFree</i>	38
35.	Resultados de los usuarios, del diez al doce, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> y <i>Ultraleap's TouchFree</i>	39
36.	Resultados de los usuarios, del trece al quince, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> y <i>Ultraleap's TouchFree</i>	40
37.	Resultados de los usuarios, del dieciséis al dieciocho, en la segunda tarea utilizando las técnicas de apuntado y clic en este mismo orden: <i>Leap Pointer</i> y <i>Ultraleap's TouchFree</i>	41
38.	Puntuación media de cada pregunta del cuestionario subjetivo.	42
39.	Puntuaciones medias de cada técnica.	43

Referencias y citas bibliográficas

- [1] Javier Pastor. “OptiKey, la aplicación de seguimiento de los ojos que plantea un futuro mejor para los enfermos de ELA” 2015. [En línea]. Disponible en: <https://www.xataka.com/medicina-y-salud/optikey-la-aplicacion-de-seguimiento-de-los-ojos-que-plantea-un-futuro-mejor-para-los-enfermos-de-ela>.
- [2] Javier Pastor. “Kinect fue «una de las mayores contribuciones» al videojuego, según Phil Spencer” 2021. [En línea]. Disponible en: https://as.com/meristation/2021/12/09/noticias/1639039638_661261.html.
- [3] Guillermo Mayoraz. “CUIDADO CON EL SÍNDROME DEL «BRAZO DE GORILA»” 2015. [En línea]. Disponible en: <https://tecnovortex.com/sindrome-del-brazo-de-gorila/>.
- [4] Jamin Hu. “Initial comparison of different hand tracking keyboard input methods for AR” 2021. [En línea]. Disponible en: https://twitter.com/jamin_hu/status/1465252009041440771. 2021.
- [5] ONU. “Objetivo 3: Garantizar una vida sana y promover el bienestar para todos en todas las edades” [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/health/>. [Accedido:20-dic-2022].
- [6] Nicole Marie Richardson. “One Giant Leap for Mankind” 2013. [En línea]. Disponible en: <https://www.inc.com/30under30/nicole-marie-richardson/leap-motion-david-holz-michael-buckwald-2013.html>.
- [7] Tom Foster. “Will These Guys Kill The Computer Interface As We Know it” 2013. [En línea]. Disponible en: <https://www.popsi.com/technology/article/2013-07/will-these-guys-kill-computer-interface-we-know-it/>.
- [8] Alexia Tsotsis. “OcuSpec Raises 1.3M From Andreessen And Others To Build An «Affordable Kinect»” 2011. [En línea]. Disponible en: <https://techcrunch.com/2011/06/10/ocuspec-raises-1-3m-from-andreessen-and-others-to-build-an-affordable-kinect/>.
- [9] Matthew Panzarino. “Leap Motion launches Software Developer Program and starts sending test units of its 3D controller” 2012. [En línea]. Disponible en: <https://thenextweb.com/news/leap-motion-launches-software-developer-program-and-starts-sending-test-units-of-its-3d-controller>.
- [10] Darrell Etherington. “Leap Motion Controller Ship Date Delayed Until July 22, Due To A Need For A Larger, Longer Beta Test” 2013. [En línea]. Disponible en: <https://techcrunch.com/2013/04/25/leap-motion-controller-ship-date-delayed-until-july-22-due-to-a-need-for-a-larger-longer-beta-test/>.
- [11] Sean Buckley. “Leap Motion’s latest motion tracking tech can see your joints” 2014. [En línea]. Disponible en: <https://www.engadget.com/2014-05-28-leap-motions-beta-tracking-tech-can-see-your-joints.html>.
- [12] Sean Buckley. “Leap Motion’s next sensor is designed specifically for virtual reality” 2014. [En línea]. Disponible en: <https://www.engadget.com/2014-08-28-leap-motion-s-next-sensor-is-designed-specifically-for-virtual-r.html>.
- [13] Ben Lang. “Orion is Leap Motion’s Overhauled VR Hand Tracking Engine” 2016. [En línea]. Disponible en: <https://www.roadtovr.com/leap-motion-launches-overhauled-hand-tracking-engine-thats-made-for-vr/>.
- [14] hackmag. “Let’s code for Leap Motion!” 2018. [En línea]. Disponible en: <https://hackmag.com/coding/lets-code-for-leap-motion/>. [Accedido:09-oct-2022].
- [15] Adi Robertson. “Leap Motion, the gesture startup reportedly almost acquired by Apple, sells to UK haptics company” 2019. [En línea]. Disponible en: <https://www.theverge.com/2019/5/30/18645604/leap-motion-vr-hand-tracking-ultrahaptics-acquisition-rumor>.

- [16] Ultraleap. “World-leading hand tracking and haptics” [En línea]. Disponible en: <https://www.ultraleap.com/company/about/>. [Accedido:18-sep-2022].
- [17] Ultraleap. “Leap Motion Controller, What’s included”. [En línea]. Disponible en: <https://www.ultraleap.com/product/leap-motion-controller/#whatsincluded>. [Accedido:16-sep-2022].
- [18] Leap Motion. [En línea]. Disponible en: <https://blog.leapmotion.com/wp-content/uploads/2014/08/Leap-Exploded-hero.png>. [Accedido:09-oct-2022].
- [19] Ultraleap. “Ultraleap for Developers” [En línea]. Disponible en: <https://developer.leapmotion.com/>. [Accedido:12-oct-2022].
- [20] Ultraleap. “Leap Motion Controller Datasheet” [En línea]. Disponible en: https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf. [Accedido:11-nov-2022].
- [21] Ultraleap. “ULTRALEAP PLUGIN FOR UNITY” [En línea]. Disponible en: <https://developer.leapmotion.com/unity>. [Accedido:11-nov-2022].
- [22] Ultraleap. “CONFIGURE TOUCHFREE”. [En línea]. Disponible en: <https://images.squarespace-cdn.com/content/v1/606cb0962166460365da1260/b9f45a62-5e42-42dd-8dc0-5cb8dceb895a/camera-position-frame.png?format=750w>. [Accedido:14-sep-2022].