



Técnicas avanzadas de HCI con uso de rastreadores de ojos y manos

Trabajo práctico especial

Integrantes:

- Sal-lari, Julieta (49629)
- Perez Cuñarro, Javier (49729)

Fecha de entrega: 10 de junio de 2015

Objetivo

El objetivo de este trabajo es poner en práctica los conocimientos adquiridos sobre técnicas de rastreo de manos. Esto comprende su rol dentro de la interacción entre el hombre y la computadora, y las buenas prácticas en cuanto a la experiencia del usuario.

El trabajo consiste en crear una aplicación pensada íntegramente para usar con Leap Motion. Debe poder usarse mediante la detección de gestos con las manos y contar con dos o más funcionalidades.

Guía para su utilización

Dado que la aplicación es un archivo .html, simplemente se debe acceder a él a través de cualquier navegador web. Su contenido se adapta al tamaño de la ventana, permitiendo su uso en pantallas de distinto tipo.

El juego consiste en ordenar un rompecabezas básico, pensado para niños. Del lado derecho de la pantalla se muestran las piezas desordenadas, y el objetivo es ordenarlas llevándolas del lado izquierdo de la pantalla hasta lograr la imagen completa.

El usuario debe señalar la ficha que quiera posicionar en el tablero y llevarla con el movimiento del dedo hasta el lugar en la izquierda que le corresponde a esa ficha. Esto se debe hacer con todas las fichas.

Una vez ordenado el rompecabezas, aparece un mensaje que felicita al usuario y se pasa al nivel siguiente. También se puede cambiar de nivel en cualquier momento con el gesto que se conoce como "swipe": hacia la izquierda pasa al siguiente nivel y hacia la derecha vuelve al nivel anterior. El "swipe" requiere del uso de varios dedos de la mano.

En su estado actual, la aplicación cuenta con cuatro niveles. Cada nivel cuenta con una ilustración distinta dividida en 4 piezas de igual tamaño y forma.

Se pueden agregar niveles extra agregando un archivo .jpg a la carpeta /images/puzzles/N.jpg, donde N es el número de nivel próximo a los que ya existen. Este archivo debe ser una imagen de cualquier tamaño, sin importar su proporción. Para que la aplicación reconozca este nivel, hay que actualizar el valor de la variable "levelQty" en las primeras líneas del código javascript.

Sobre la experiencia de usuario

El trabajo comenzó con un estudio de los usuarios finales de la aplicación: los niños. Haciendo uso de la técnica de "Personas" (tal como se la entiende en el mundo UX), se creó un personaje imaginario para representar al usuario que terminaría jugando con la aplicación. Este niño imaginario sirvió mucho al definir cómo sería la aplicación. Sirvió para el análisis de las capacidades del usuario, los gestos que conoce y usa, su forma de pensar, su objetivo al usar la aplicación y el contexto en el cual la usaría.

De esta forma, se llegó a la conclusión de que la manera más fácil de mover piezas sería apuntando con el dedo, en lugar de hacer que "haga click" sobre la misma o que intente agarrarla cerrando la mano. Es la forma en la que los chicos más chicos tratan de hacer cosas, por lo que un juego pensado para ellos debe respetarlo y no forzarlos a repensar sus impulsos. Después de todo, lo que se busca es que encuentren la forma de jugar sin leer un manual o recurrir a una ayuda.

También se decidió por el uso de un cursor permanente, con aspecto amigable y feedback constante (señalando el estado actual, si está libre o moviendo una ficha). El paso al siguiente nivel se hizo automático para sacarle una carga al usuario, es decir, que no tenga que adivinar cómo seguir jugando, frustrarse al no saber cómo y terminar abandonando la aplicación.

Sobre el proceso de programación

Si bien se analizaron distintas maneras de codificarlo, se decidió hacer el código en HTML y Javascript, interactuando con el Leap Motion mediante el framework LeapJS visto en clase. Esto permite tener una forma sencilla de ejecución y distribución vía internet.

Recapitulando lo mencionado en la guía de utilización, se eligieron como gestos los movimientos de swipe y señalización con un dedo. Para la señalización se optó por dividir la pantalla en zonas para reconocer si se posicionó sobre una ficha o sobre el tablero. La obtención de la posición actual se logró normalizando las coordenadas recibidas del interaction box al espacio brindado por el navegador web. La señalización se limitó al uso de un único dedo extendido a la vez, mientras que el swipe requiere de 3 o más dedos simultáneos.

Un desafío que se enfrentó a la hora de programar el swipe fue el reconocimiento continuo de los gestos por parte de la API de LeapJS. Los gestos que trabaja nativamente no se reconocen una única vez, sino que se detectan frame tras frame hasta que el gesto concluye. De no estar atento, el código propio podría terminar reaccionando a un gesto una decena de veces. Por este motivo, se hizo uso de la función `_.debounce` de Underscore.JS para evitar la ejecución de `swipe()` varias veces seguidas.