

Software de comunicación para personas con parálisis cerebral

Javier García González-Albo



Trabajo de Fin de Grado del Grado de Software Facultad de
Informática
Universidad Complutense de Madrid

Director: Luís Garmendia Salvador.
Codirector: Asunción Reyes Corrochano.

Apartado	Página
1. Índice	1
2. Introducción	3
3. Estado del arte	5
4. Programas probados	9
4.1. Preámbulo	9
4.2. TrackEye	9
4.3. FaceTrackNoIR	12
4.4. QMouse	13
4.5. HeadMouse	14
4.6. Licencia HeadMouse	14
5. Manual de Instalación	17
6. Manual de uso	19
6.1. HeadMouse	19
6.2. EyeSpeaker	21
6.3. ClickSpeak	23
7. Funcionamiento	27
8. Pruebas	33
9. Plan de Trabajo	35
10. Trabajos futuros	37
11. Conclusiones	39
12. Bibliografía	41

2.1. Objetivo y descripción del proyecto

El objetivo de este Trabajo de Fin de Grado es la creación de un software para la comunicación de personas con discapacidad exclusivamente motriz, incluyendo el habla, pero sin tener afectada la cognición; concretamente va dirigido a un niño que padece una parálisis cerebral. Dicho software se ha desarrollado en dos programas separados.

El primero es un software de seguimiento de pupila basado en la tecnología de EyeTracking, para lo que se usa un ratón de mirada proporcionado por un programa ya existente. La aportación en este apartado ha sido crear una interfaz, como capa superior, mediante botones que ejecutan las acciones correspondientes a seleccionar el método de comunicación y las opciones correspondientes a cada método que ofrece el sistema.

Nuestro niño, Moisés, sufre disfonía, lo que implica fluctuación en el tono muscular, unas veces hipertónico y otras hipotónico en distintas partes de su cuerpo, afectándole a la sujeción de la cabeza. Por ese motivo se ha desarrollado un segundo programa que funciona con ratón y teclado, al cual se le ha añadido un pulsador. Incluye un menú para seleccionar distintas formas de comunicación. Las opciones de este menú se le ofrecen al usuario de forma secuencial ya que solo podrá comunicarse con el sistema mediante clicks de ratón o pulsaciones de teclado.

En España la empresa Vía Libre de la fundación ONCE vende materiales ortopédicos y de comunicación, todos con licencia de uso, como el usado por Stephen Hawking, no asequibles a familias que ya tienen gastos en sillas, adaptadores, y material ortopédico.

Este segundo programa ha sido probado y modificado según las principales necesidades de comunicación de Moisés, es de código abierto y puede ser descargable desde <https://sourceforge.net/p/clickspeak/>

2.2. Palabras de búsqueda

- Seguimiento de pupila
- Comunicación
- Parálisis cerebral
- Formularios de Windows
- Temporizador

2.3. Objective and description of the project

The objective of this Grade Project is the creation of communication software for exclusively motor disabled people, included speech, without affected cognition: Specifically directed for a cerebral palsy affected child, said software has been developed in two separated programs.

The first one is a Eye Tracking system based on EyeTracking tecnology, which uses an eyetracking mouse provided by an already existing program. The aportation in this section is the implementation of an interface as a superior layer, through buttons that will execute the corresponding orders, selecting the communication chanel and the subsequent options following each chanel offered by the system.

The subject child, Moisés, suffers of disponia, wich implies fluctuating muscle tone, hipertonic or hipotonic in different parts of his body alternating, affecting head stability. Because of that has been created a second, mouse and keyboard functioning program, with an added pulsator. It includes a selecting menu, for the different communication means. The options in this menu are offered to the user secuentially, given the limit of communication by clicks or key pressing.

In Spain, ONCE's Vía Libre enterprise sells ortopedic and communication articles, all licensed, like the one used by Stephen Hawking. These are not affordable for families that are already spending on chairs and othopedic material.

This second program has already been tested and modified according to the main communication needs of the subject, Moisés, is free code, and is avaiable for download from <https://sourceforge.net/p/clickspeak/>

2.4. Key Words

- Eye Tracking
- Communication
- Cerebral palsy
- Windows Forms
- Timer

Podemos considerar el EyeTracking [1] como una tecnología que permite seguir los movimientos oculares de una persona para inferir qué mira y qué ve.

Existen diversos sistemas para determinar el movimiento de los ojos. El más común es a través de la luz, por lo general luz infrarroja, que se refleja en los ojos y se capta mediante una cámara de video o algún otro sensor óptico.

En cuanto al software y equipos existentes de EyeTracking nos encontramos con tres tipos de sistemas en la actualidad:

1. Equipos montados en la cabeza: El usuario debe llevar el equipo en la cabeza durante toda la sesión, siendo en algunos casos molesto y un tanto intrusivo.
2. Gafas de EyeTracking: Incorporan la tecnología de seguimiento de los ojos, incorporando tanto la cámara como las luces infrarrojas. Dichas gafas van obteniendo la información y transmitiéndola mediante wifi o bluetooth al equipo que almacena la información.
3. Movimiento ocular en la distancia: Utilizando cámaras de alta velocidad que se encargan de rastrear los movimientos de los globos oculares, dilatación pupilar y parpadeo del sujeto. La información obtenida sirve para crear los "puntos calientes", es decir, aquellos puntos donde la vista se detiene más tiempo además de determinar los recorridos visuales.

Para este proyecto se utilizará el último método de los citados, al ejecutar un programa externo que permite utilizar el ratón con los ojos y una cámara web convencional.

Se pueden distinguir dos vertientes de aplicación de sistemas de EyeTracking:

1. Diagnósticas: el sistema de EyeTracking proporciona la evidencia objetiva y cuantitativa de los procesos relacionados con la visión y la percepción.
2. Interactivas: el sistema de EyeTracking sirve como un dispositivo de entrada sustituyendo al ratón.

La vertiente interactiva tiene numerosas aplicaciones prácticas, tales como su uso en entornos de realidad virtual o por usuarios con discapacidad motriz. Esta última aplicación es en la que se centra este trabajo como se ha mencionado anteriormente.

Recientemente se está utilizando la tecnología de EyeTracking en el campo de la usabilidad y el diseño de páginas Web. La idea es analizar las zonas de la pantalla a las que está mirando un sujeto y de esta forma mejorar el diseño y la usabilidad de la página. Para realizar esta tarea se utilizan dos técnicas:

1. Mapa de puntos calientes: con el Mapa de puntos calientes se obtiene una imagen, en distintos colores, de lo que está viendo el sujeto. Las zonas con colores fríos indican que el sujeto ha mirado poco ahí mientras que los colores cálidos indican las zonas en las que el sujeto ha mirado durante más tiempo. [2]

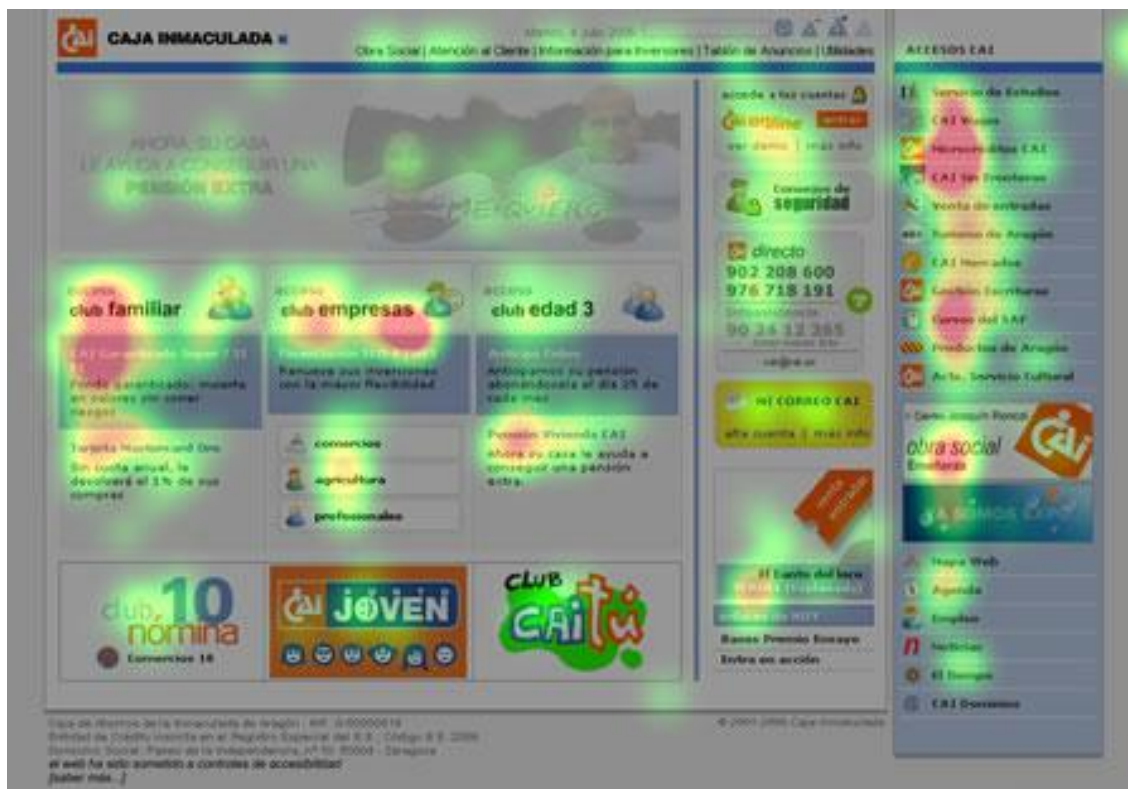


Ilustración 1 - Mapa de puntos calientes

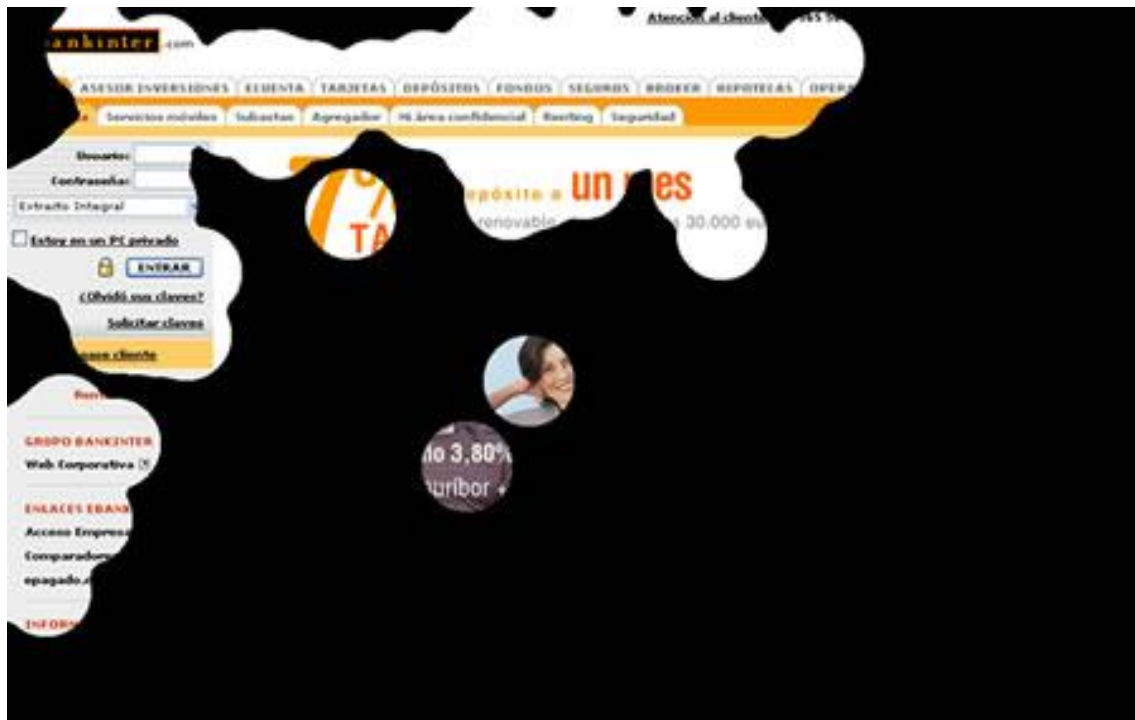


Ilustración 2 - Mapa de zonas ciegas

2. Mapa de zonas ciegas: con el Mapa de zonas ciegas se crea una imagen en la que solo se ven las regiones a las que ha mirado el sujeto mientras que el resto de la imagen está completamente en negro.

Otro campo en el que se utiliza la tecnología de EyeTracking es el que se conoce como Neuromarketing [3]; que consiste en la aplicación de técnicas pertenecientes a las neurociencia aplicadas al ámbito de la mercadotecnia, cuyo objetivo es entender los motivos que llevan a un consumidor a tomar unas decisiones u otras a la hora de comprar diferentes productos.

La información que recogen los sistemas de seguimiento visual pueden servir para conocer los recorridos visuales de los sujetos y crear mapas que señalen los puntos “calientes” de la imagen. Al igual que se ha mencionado en el ámbito de la usabilidad y el diseño de páginas web.

Otras utilidades de la tecnología EyeTracking son: Sistemas avanzados de Asistencia a la Conducción, Evaluación de la visibilidad en un puesto de trabajo, Detección de la mentira, Estudios de ergonomía, Diagnósticos clínicos, Diseño de páginas Web, Marketing, etc.

Actualmente, en el mercado existen varios programas que implementa un ratón de mirada. El más conocido se llama "Tobii PCEye" y ha sido creado por la compañía Tobii Technology. [4]

Tobii Technology es una compañía multinacional de origen sueco, afincada en Estocolmo, de alta tecnología que desarrolla y vende productos de EyeTracking. En 2008, ganó el the Grand Award of Design, que se entrega a las compañías de ingeniería suecas; por sus pantallas controladas con los ojos. Sus productos se venden a todo el mundo tanto directamente como a través de distribuidores. La empresa salió a bolsa en su país de origen en abril de 2015.

Tobii PCEye puede ser instalado en cualquier ordenador personal con sistema operativo Windows. Es un dispositivo que detecta el movimiento que realiza el usuario con los ojos, desplazando el puntero hacia el lugar deseado en el monitor. Operativamente, el usuario sólo tiene que dirigir la mirada hacia el punto de la pantalla en donde quiere que se posicione el puntero. Para realizar las funciones de los clic de los botones de ratón, el producto suele proporcionar un software que emula dichas funciones.

4.1. Preámbulo

Crear un ratón de mirada no es una tarea sencilla; por ello, el primer paso antes de empezar a programar ha sido buscar aplicaciones ya existentes que implementen las funcionalidades buscadas y utilizarlas en este proyecto si es posible.

Como se ha mencionado anteriormente, la descripción de este trabajo es la creación de un software dirigido a personas con movilidad reducida con el objetivo de que dichas personas puedan comunicarse mediante imágenes y la escritura manejando el sistema con la vista. Por lo que el ratón de mirada es un medio para la realización de dicho trabajo pero no es el fin del mismo.

En la red he encontrado varios programas que parecen cumplir los requisitos que se necesitan; algunos son de código abierto y de otros solo se ha podido obtener el ejecutable. A continuación se desarrollará el funcionamiento y características de cada uno de ellos y se analizarán las ventajas e inconvenientes que tienen.

4.2. Track Eye

TrackEye [5] es un programa de código abierto desarrollada en C++ por un programador de origen turco cuyo nombre es Zafer Savas. Los objetivos de esta aplicación son:

- Seguimiento en tiempo real de la cara con escala y rotación invariante.
- Seguimiento de las áreas de ojos de forma individual.
- Seguimiento de las características oculares.
- Hallar la dirección de la mirada.
- Control remoto mediante movimientos oculares.

Este programa utiliza la biblioteca OpenCV de C++ para usar las WebCam y tratar las imágenes capturadas por estas. La implementación del seguimiento del ojo se realiza en tres procesos diferentes:

1. Detección facial.
2. Detección ocular.
3. Extracción de características oculares.

Detección facial

Se utilizan dos métodos diferentes en este proceso del proyecto:

1. Continuously Adaptive Means-Shift Algorithm.
2. Método de detección facial Haar.

Continuously Adaptive Means-Shift Algorithm

Adaptive Mean Shift algorithm se utiliza para el seguimiento facial humano y se basa principalmente en la técnica no paramétrica para escalar gradientes de densidad con el objetivo de encontrar el pico de las distribuciones de probabilidad llamando al algoritmo "Mean Shift Algorithm". Como el seguimiento facial se realiza en secuencias de vídeo, el "Mean Shift Algorithm" se modifica para tratar el problema de cambiar dinámicamente las distribuciones de probabilidad del color. El diagrama de bloques del algoritmo se da a continuación:

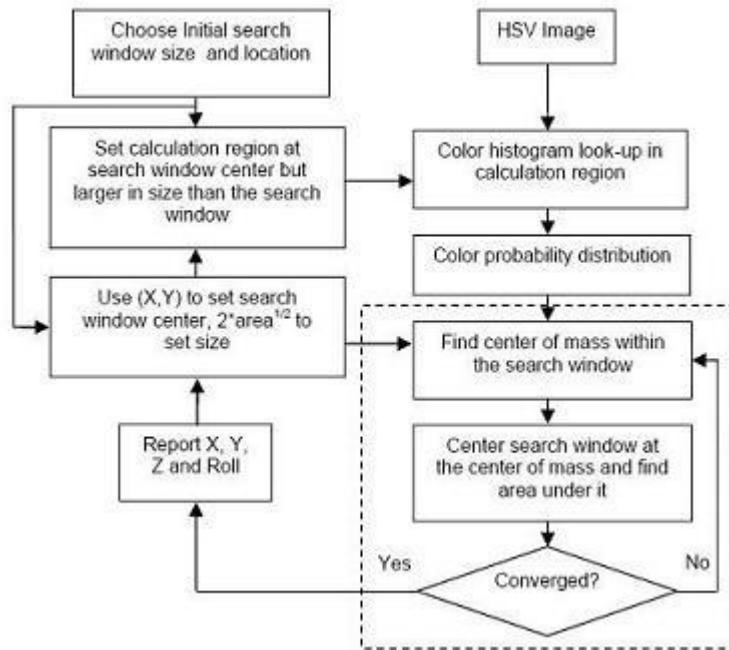


Ilustración 3 - Algoritmo Means-Shift

Método de detección facial Haar

El segundo algoritmo de detección facial se basa en un clasificador que trabaja con características Haar. En primer lugar, se entrenó con unos pocos cientos de muestras faciales. Una vez que el clasificador está entrenado, se puede aplicar a una región de interés en una imagen de entrada. La salida del clasificador es un "1" si en la región es probable que se muestre la cara y "0" en caso contrario. Para buscar el objeto en toda la imagen, se puede mover la ventana de búsqueda a través de esta y comprobar cada ubicación utilizando el clasificador. El clasificador está diseñado de modo que se pueda fácilmente cambiar de tamaño con el fin de ser capaz de encontrar los objetos de interés en diferentes tamaños, lo que es más eficiente que el cambio de tamaño de la imagen en sí.

DetECCIÓN OCULAR

También se utilizan dos métodos diferentes:

1. Comparación de plantillas.
2. Procedimiento Adaptive EigenEye.

Comparación de plantillas

La comparación de plantillas es un método bien conocido para la detección de objetos. En el método de correspondencia de plantillas usado en este programa, se crea un patrón de ojo estándar manualmente y se le da una imagen de entrada, los valores de correlación con los patrones estándar se calculan para los ojos. La existencia de un ojo se determina basándose en los valores de correlación. Este enfoque tiene la ventaja de ser simple de implementar. Sin embargo, a veces puede ser inadecuado para la detección ocular ya que no puede tratar eficazmente con la variación en escala, pose y forma.

Procedimiento Adaptive EigenEye

La idea principal del método Adaptive EigenEye es descomponer imágenes del ojo en un pequeño conjunto de características llamadas eigeneyes, que pueden ser considerados como los principales componentes de las imágenes originales. Estos eigeneyes funcionan como los vectores de la base ortogonal de un subespacio llamado eyespace.

La ventaja de esta aplicación es que es de código abierto y programada en C++, lenguaje que más domino hasta el momento; pero tiene grandes inconvenientes por los cuales he desechado utilizarlo en mi proyecto.

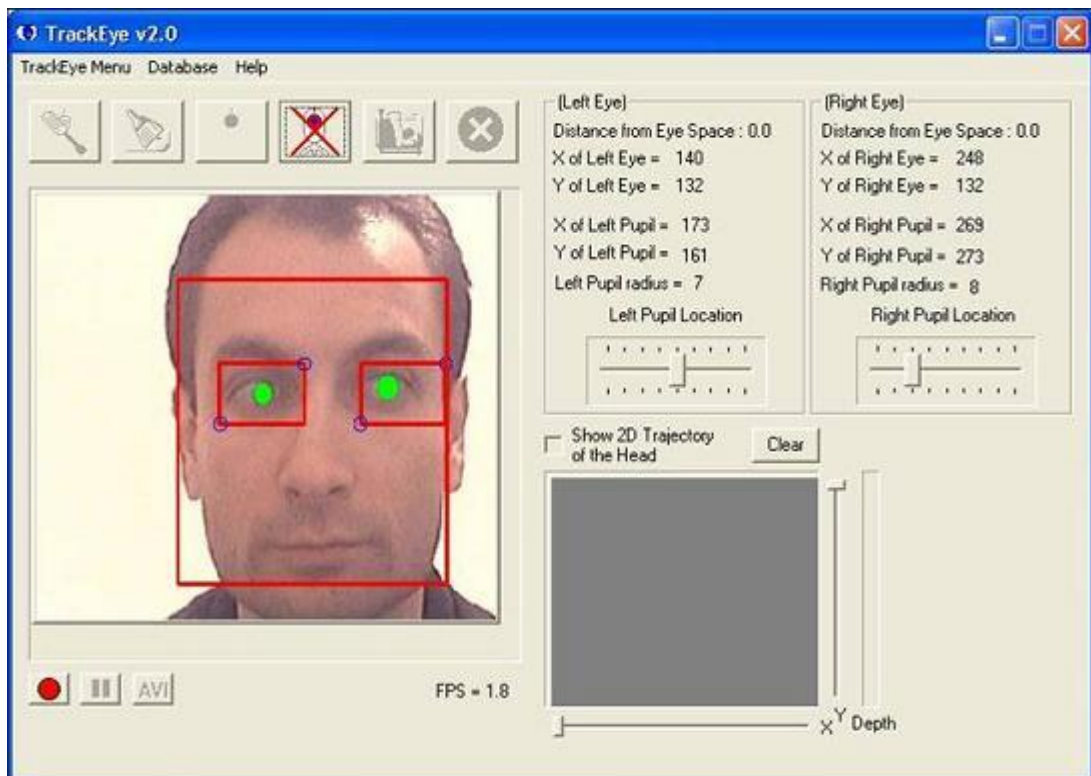


Ilustración 4 - TrackEye

La versión ejecutable no es compatible con mi cámara web que viene de serie en el portátil. Además, el código fuente proporcionado por el creador no funciona correctamente por dos motivos; se necesitan archivos de la biblioteca OpenCV, pero una vez incluidos en el proyecto, encontré declaraciones de variables cuyos tipos no estaban definidos en ningún archivo.

4.3. FaceTrack NoIR

FaceTrackNoIR [6] es un software utilizado en el mundo de los videojuegos para cambiar los controles típicos de mouse, teclado y/o joystick por movimientos de cabeza para manejar los personajes de los juegos.

Su funcionamiento es el siguiente: una cámara web envía un flujo de vídeo al ordenador, proporcionando así una entrada masiva de datos en bruto. Después, un sofisticado software filtra las imágenes de video y localiza puntos de referencia (generalmente luces IR brillantes). Mediante estas referencias, el software calcula el movimiento y la posición en a relación a un centro artificial. Ahora, estos datos se puede utilizar para mover la cabeza en el juego. Es una forma muy natural de movimiento en comparación con el joystick y los botones.

La belleza de FaceTrackNoIR reside en que el software de seguimiento emplea puntos de referencia faciales para hacer los cálculos, por lo que no se requieren construcciones LED.

FaceTrackNoIR se basa en la teoría de los seis grados de libertad para el movimiento de los personajes a partir de los movimientos de cabeza correspondientes.

Este software está programado en C++, con Visual Studio. Su licencia es de código abierto por lo que se puede acceder a su código fuente. Esta es la gran ventaja de FaceTrackNoIR pero no es útil para el objetivo de mi proyecto ya que no captura la posición de los ojos. Además, al igual que el programa mencionado anteriormente, el código fuente no está completo en el directorio de SourceForge.



Ilustración 5 - FaceTrackNoIR

4.4. Q-Mouse

Otro sistema que permite el control del ratón con la vista es Q-MOUSE [7]. Q-MOUSE es una aplicación que permite sustituir el mouse y el teclado, dando lugar a personas con discapacidad a utilizar un PC de forma sencilla. Mediante una webcam se detectan los movimientos del rostro y se envían al puntero del mouse para controlarlo. Realizando ciertos gestos faciales se simulan los clicks del mouse. Incluye también un teclado virtual para simular el uso del teclado real, con predicción de palabras para agilizar la escritura.

Su funcionamiento es el siguiente: Con el uso de una webcam se detectan todos los movimientos de la cara, dependiendo de los movimientos se mueve el puntero del ratón de un lugar a otro. Para hacer clic con el ratón, se debe determinar de antemano un movimiento particular de la cara; QMouse además incluye un teclado virtual para que, a través de ciertos gestos, se pueda escribir.

No se ha podido comprobar el funcionamiento de este programa porque no es compatible con la Web Cam que viene de serie en el portátil así que se ha desechado utilizarlo en el trabajo.



4.5. HeadMouse

[8] HeadMouse es un programa gratuito diseñado para sustituir el ratón convencional, permite controlar el desplazamiento del cursor con pequeños movimientos de la cabeza y realizar acciones de clic mediante gestos faciales realizados delante de una webcam.

HeadMouse ha sido creado por la Cátedra de Accesibilidad de la Universidad de Lleida y se ofrecen de forma gratuita gracias a Indra, Fundación Adecco y Universidad de Lleida.

La idea inicial de HeadMouse surgió de la posibilidad de aplicar algoritmos de visión artificial utilizados en robótica para controlar el desplazamiento del cursor en la pantalla del ordenador mediante el desplazamiento de la cabeza.

El programa ejecutable es gratuito pero el código no es libre por lo que se necesita tener instalado dicho programa en el ordenador ya que mi proyecto realiza una llamada a su ejecutable al arrancar.

4.6. Licencia HeadMouse

Licencia de uso

INFORMACION AL USUARIO: Lea atentamente el siguiente acuerdo de Licencia de Uso. La instalación o uso del software proporcionado con este acuerdo supone que Usted acepta los términos del mismo.

1. CONTENIDO DE LA LICENCIA

El usuario obtendrá el software y la documentación con el derecho no transferible y no exclusivo de uso. El alcance de este derecho queda exclusivamente determinado por los términos y condiciones establecidos en este acuerdo. Cualquier otro uso o explotación quedan excluidos. Si alguna de las cláusulas de esta licencia no se ajusta a la legalidad quedará anulada pero el resto de la licencia mantendrá su efecto.

2. CONCESION DE LICENCIA

Según los términos de este acuerdo, se le otorga a Usted una licencia no exclusiva y no transferible que le permitirá usar y reproducir el programa para uso personal o interno, siempre y cuando todas y cada una de las copias realizadas contengan todos los avisos de propiedad sin modificaciones, incluyendo este acuerdo de licencia. Esta licencia no incluye el derecho a recibir ningún soporte técnico, actualizaciones o cualquier otro servicio.

3. PROPIEDAD INTELECTUAL

Tanto el programa, como los correspondientes derechos de propiedad industrial o intelectual son propiedad del Grupo de Robótica de la Universidad de Lleida y están protegidos por las leyes españolas de Propiedad Intelectual y por las disposiciones de los tratados internacionales que sean de aplicación.

4. RESTRICCIONES

Usted reconoce y acuerda que no podrá: (a) modificar o crear ningún producto derivado del programa o sus documentos; (b) intentar descompilar, desensamblar, realizar ingeniería inversa o intentar derivar el código fuente del programa;

(c) vender, arrendar o alquilar ninguna parte del programa a un tercero; o (d) retirar o alterar cualquier marca registrada, logotipo o cualquier otro aviso, leyenda, símbolo o etiqueta de propiedad contenida en el programa o producto.

5. GARANTÍA LIMITADA

Este programa y la documentación relacionada se proporcionan "tal cual" sin garantía ni condición de ningún tipo, expresa o implícita, incluyendo, pero sin limitarse a, las garantías y condiciones de comerciabilidad, idoneidad para un determinado fin y ausencia de infracción. Usted asume todo riesgo que surja de la utilización o del rendimiento de este programa.

6. LIMITACIÓN DE RESPONSABILIDAD.

El licenciatarario será responsable de cualquier daño o pérdida de utilidad del software y asume totalmente el riesgo de instalar y probar el software.

El propietario del software, al amparo de la ley, no ofrece ninguna otra garantía, ni explícita ni implícita, ni estatutaria ni ninguna otra, respecto al manual y al software del programa y en ningún caso será responsable de pérdidas directas o indirectas, incidentales o consecuenciales, que resulten de cualquier defecto del software, incluso cuando se haya avisado de la posibilidad de dichas pérdidas. El propietario del software no se hará responsable de los costos de los servicios de reparación que deban ser efectuados por el uso del software.

Mi trabajo no es un "producto" ya que nunca se intentará comerciar con él, tampoco se ha aplicado ingeniería inversa para acceder a su código, simplemente se utiliza su ejecutable; por lo que no viola ninguna de las cláusulas de la licencia de HeadMouse.

A continuación se explican los pasos a seguir para la instalación de los programas de producción propia y el ratón de mirada.

5.1. HeadMouse

1. Ejecutar el archivo "headmouse2.exe".
2. Seleccionar el idioma del instalador.
3. Cerrar las aplicaciones abiertas en el PC antes de continuar y pulsar siguiente.
4. Acuerdo de licencia:
 - a. Leer el acuerdo de licencia.
 - b. Aceptar el acuerdo de licencia
 - c. Pulsar en Siguiente.
5. Seleccionar la carpeta en la que se guardará el programa y pulsar en Siguiente.
6. Seleccionar la carpeta en la que se creará el acceso directo en caso de querer hacerlo o seleccionar la opción "No crear una carpeta en el Menú Inicio".
7. Seleccionar si se quiere crear un icono en el escritorio.
8. Esperar a que se instale el programa.
9. Seleccionar, si se desea, ejecutar el programa tras terminar la instalación.

5.2. EyeSpeaker

1. Ejecutar el archivo "EyeSpeakerSetup".
2. Pulsar en Siguiente.
3. Seleccionar la carpeta en la que se instalará la aplicación y los usuarios que podrán utilizarlo.
4. Pulsar en Siguiente.
5. Esperar a que se instale la aplicación.
6. Pulsar en Cerrar.

5.3. Clickpeak

Seguir los mismos pasos que para EyeSpeaker ejecutando el archivo ClickSpeak.

A continuación se exponen una serie de tutoriales para conocer el funcionamiento del ratón de mirada y los programas de producción propia.

6.1. HeadMouse

Antes de ejecutar la aplicación EyeSpeaker se debe configurar el HeadMouse; para ello se debe ejecutar el programa HeadMouse primero. En caso de haber más de una cámara conectada al equipo habrá que seleccionar la fuente de imagen deseada.

Al hacerlo aparecerá la cara del usuario en pantalla y el programa realizará automáticamente la captura de la cabeza y los ojos; a continuación se podrá mover el puntero del ratón por la pantalla moviendo solo los ojos.

Lo que nos interesa para nuestro programa EyeSpeaker es configurar las opciones de click del ratón, cosa que deberá hacer una persona que pueda usar el ratón convencional periférico. Para ello debemos mover el ratón hasta la esquina inferior derecha y esperar a que se despliegue un pequeño menú vertical. Después hay que seleccionar la opción de configuración del ratón.



Ilustración 6 - Menú HeadMouse

Finalmente, se elige la forma de hacer el click izquierdo, doble click y el resto de opciones que nos ofrece el programa.

Es muy recomendable que el click izquierdo tenga un temporizador de unos tres segundos, de esta manera nos aseguraremos de que podemos usar la aplicación en el caso de que no se detecten correctamente los gestos de parpadeo o abrir y cerrar la boca.



Ilustración 7 - Configuración HeadMouse

Una vez que se ha elegido las opciones del ratón, pulsamos en "Aceptar" y a continuación cerramos el programa eligiendo la opción correspondiente del menú vertical que se ha indicado previamente.



Ilustración 8 - Menú HeadMouse (2)

6.2 EyeSpekaer

Al arrancar la aplicación EyeSpeaker, en primer lugar se ejecutara HeadMouse con la configuración que se haya elegido previamente. A continuación se podrá navegar por las diferentes opciones de la aplicación usando HeadMouse como ratón.

La aplicación tiene tres métodos para que el usuario pueda comunicarse: Pictograma, Respuestas rápidas y Escritura.



Ilustración 9 - Bienvenida EyeSpeaker

La opción Pictograma mostrará en pantalla cuatro botones con cada dibujo correspondiente a la necesidad del usuario, estas son: Comida ("Tengo hambre"), Cama ("Tengo sueño"), Retrete ("Necesito ir al baño"), Juguetes ("Me aburro"). Al seleccionar una opción se reproducirá por voz una la frase correspondiente al dibujo indicado anteriormente.

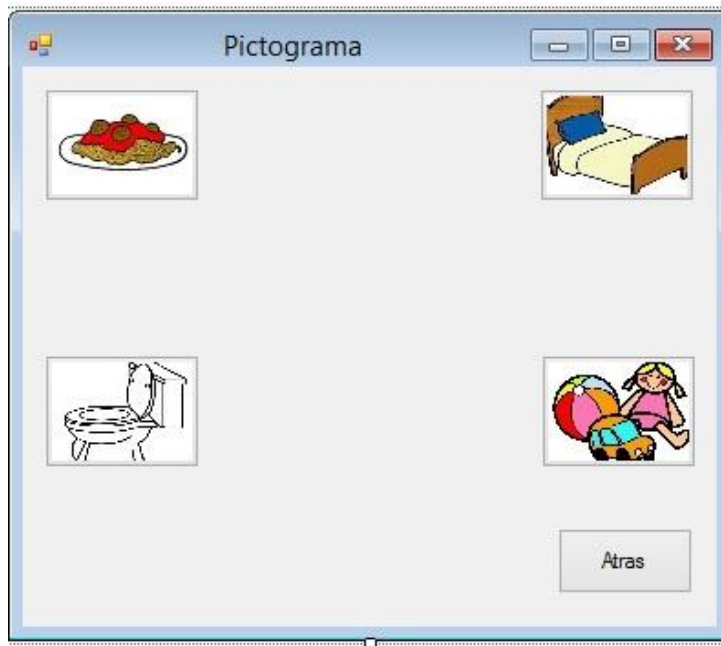


Ilustración 10 - Pictograma EyeSpeaker

La opción de Respuestas Rápidas permite al usuario contestar a preguntas de si o no eligiendo la opción correspondiente. La respuesta se reproduce mediante voz.



Ilustración 11 - Respuestas rápidas EyeSpeaker

La opción de Escritura permite al usuario escribir una frase mediante el teclado en pantalla de Windows. Cuando haya escrito lo que quiere comunicar, deberá seleccionar la opción "Mostar" para que el sistema avise de que se ha escrito un nuevo mensaje.



Ilustración 12 - Escritura EyeSpeaker

AVISO: Al volver atrás desde la opción de escritura, no se cerrará el teclado en pantalla. Para cerrarlo, el usuario deberá pinchar en la X situada en la esquina superior derecha del teclado.

Para cambiar de opción basta con pinchar el botón "Atrás" que aparece en la pantalla de cada una de las opciones y elegir la nueva opción deseada.

Para cerrar el programa se deberá pinchar el botón "Salir" de la pantalla inicial. Entonces se cerrará el HeadMouse y a continuación la aplicación.

6.3. ClickSpeak

En primer lugar se le mostrará al usuario un menú con las tres opciones disponibles de comunicarse a través del sistema. En la parte inferior de la pantalla se muestra secuencialmente y de forma cíclica las opciones indicadas en el menú anteriormente mencionado. El usuario pulsará cualquier tecla del teclado o botón del ratón cuando vea en la parte inferior la opción deseada.

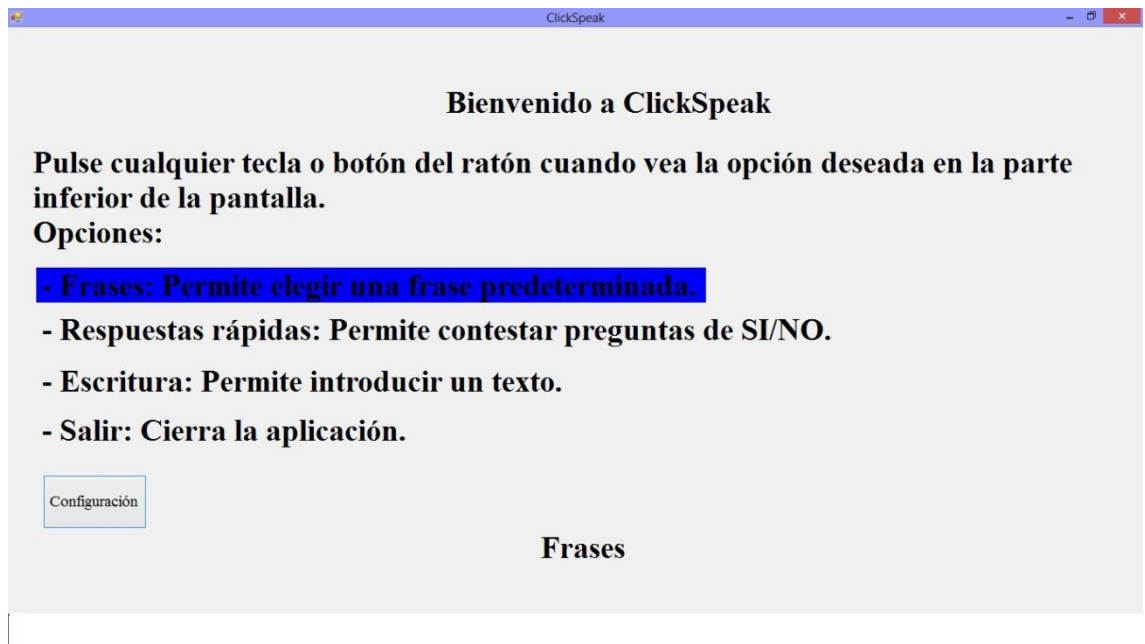


Ilustración 13 - Bienvenida ClicSpeak

La opción de Frases es equivalente al Pictograma de EyeSpeaker. Al igual que en la pantalla principal, se mostrará un menú con las diferentes opciones y secuencialmente se mostrarán en la parte inferior de la pantalla. El usuario pulsará cualquier tecla del teclado o botón del ratón cuando vea en la parte inferior la opción deseada y se reproducirá el audio correspondiente a la frase elegida.

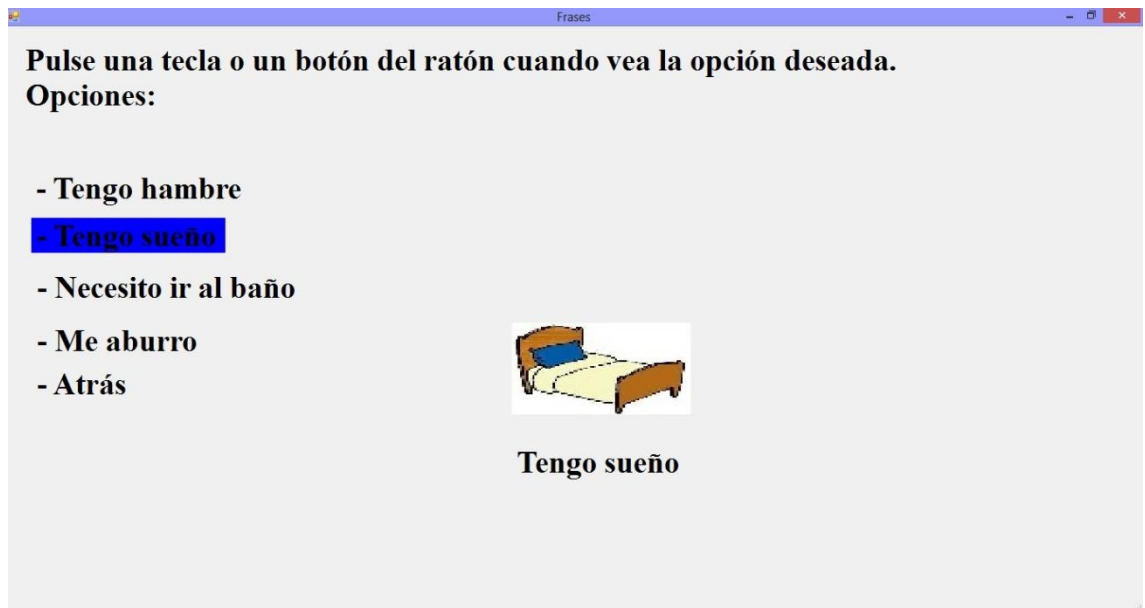


Ilustración 14 - Frases ClickSpeak

La opción de Respuestas Rápidas es análoga a la del programa EyeSpeaker y su funcionamiento es igual al de la opción Frases descrita anteriormente.

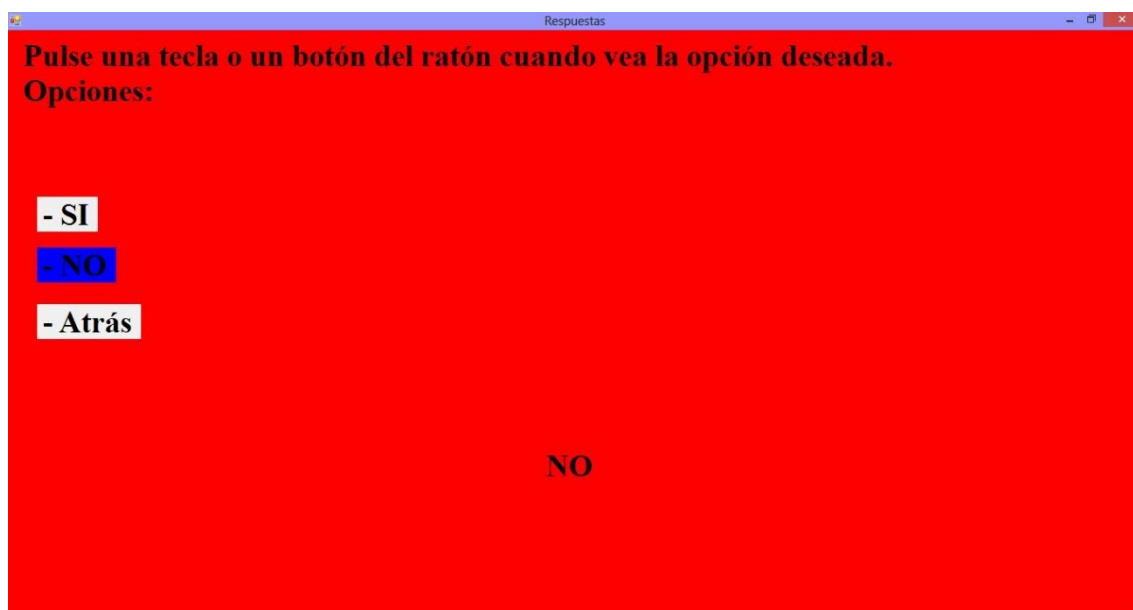


Ilustración 15 - Respuestas rápidas ClickSpeak

Finalmente, la opción de Escritura permite escribir un texto introducido por el usuario. En el centro de la pantalla se muestran las letras del alfabeto de forma secuencial, además de las opciones de Borrar una letras, introducir un Espacio y Enviar el mensaje. Para añadir una letra al texto o realizar alguna de las tres acciones mencionadas, el usuario tendrá que pulsar una tecla del teclado o botón del ratón cuando vea la letra o acción deseada en el centro de la pantalla. Al seleccionar la opción de Enviar, se mostrará el mensaje en una caja de texto y se avisará mediante la reproducción del audio que indica que se ha escrito un mensaje.

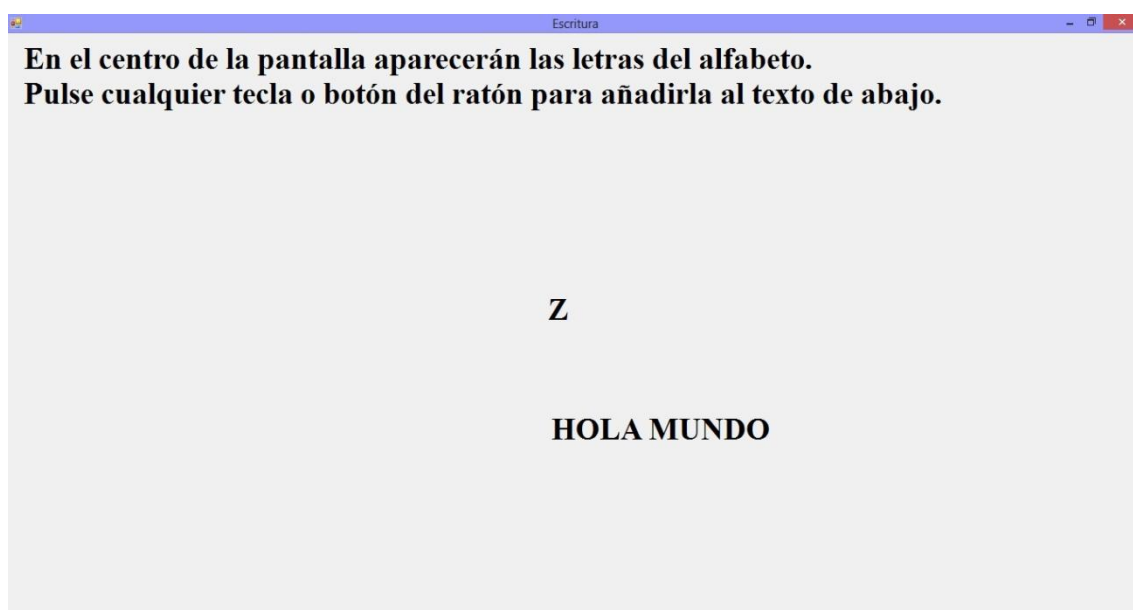


Ilustración 16 - Escritura ClickSpeak

Además se puede establecer la velocidad de recorrido de las opciones de menú entre lento, medio y rápido pinchando en el botón de Configuración que aparece en la pantalla principal del programa.

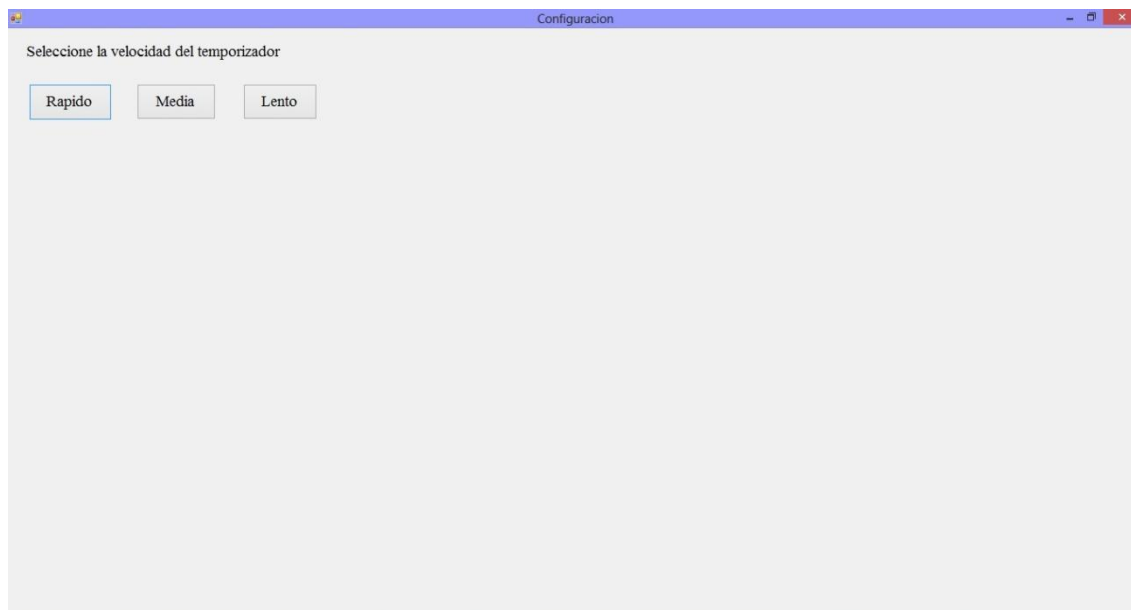


Ilustración 17 - Configuración ClickSpeak

7.1. Windows Form

Windows Forms [9], o formularios Windows, es el nombre dado a la interfaz de programación de aplicación gráfica (API) que se incluye como parte de Microsoft .NET Framework, que proporciona acceso a los elementos de la interfaz de Microsoft Windows nativas envolviendo la API de Windows existente en código administrado.

Los formularios de Windows Form se pueden crear en dos lenguajes, Visual Basic o Visual C#. Para esta aplicación he usado Visual C# utilizando el tutorial online que proporciona Microsoft de forma gratuita.

7.2. Visual C#

C# es un lenguaje de programación diseñado para compilar aplicaciones que se ejecutan en .NET Framework. Se trata de lenguajes eficaces, que presentan seguridad de tipos y están orientados a objetos. Permiten a los desarrolladores de software crear aplicaciones Windows, web y móviles.

7.3. C++

C++ es un lenguaje de programación que extiende de C, aportando mecanismos que permiten la manipulación de objetos. Es un lenguaje multiparadigma por lo que permite tanto programación estructurada como orientación a objetos.

Para crear interfaces gráficas se utilizan formularios de Windows Form que ejecutan una serie de instrucciones cuando ocurre un evento determinado.

7.4. EyeSpeaker

La aplicación creada para este trabajo ha sido construida con los formularios de Windows Form, basados en Visual C#, con el entorno de desarrollo Visual Studio 2010.

Se han creado cuatro formularios que sirven de capa superior para que el usuario pueda comunicarse a través del ordenador mientras se ejecuta al mismo tiempo el programa "HeadMouse". Estos cuatro formularios están denominados de la siguiente forma:

- Bienvenida
- Pictograma
- RespuestasRapidas
- Escritura

7.4.1. Bienvenida

El formulario Bienvenida es la primera ventana que se le ofrece al usuario al ejecutar la aplicación. Esta ventana hace una llamada al sistema para arrancar el ratón de mirada, por ello es necesario tenerlo previamente instalado en el sistema.

Este formulario consta de cuatro botones, los tres correspondientes a las opciones de comunicación tienen un código muy parecido:

En primer lugar se crea el formulario correspondiente a la opción elegida, a continuación se "esconde" la pantalla de bienvenida y se muestra el formulario que se acaba de crear. Cuando este se cierra, se vuelve a mostrar el formulario de bienvenida.

El botón Salir invoca un script de windows que cierra el ratón de mirada y posteriormente, cierra completamente la aplicación EyeSpeaker.

7.4.2. Pictograma

El formulario Pictograma consta de cinco botones, cuatro de ellos reproducen los archivos de audio correspondientes al dibujo y el último cierra el formulario.

Para reproducir los archivos de audio se obtiene en primer lugar la ruta en la que se encuentra la aplicación, y por tanto los archivos de audio. A continuación se crea un objeto de tipo System.Media.SoundPlayer con la ruta del audio para después ejecutar el método Play sobre el objeto que se acaba de crear.

El botón Atrás simplemente cierra el formulario.

7.4.3. Respuestas Rápidas

Su funcionamiento es análogo al del formulario Pictograma. Los botones SI y NO reproducen los archivos de audio correspondientes mediante objetos de tipo System.Media.SoundPlayer y el método Play. El botón Atrás cierra el formulario.

7.4.4. Escritura

Al mostrar el formulario se invoca al proceso osk, que corresponde al teclado en pantalla de windows. Una vez que se introduce el texto en la textbox y se pulsa el botón de Mostrar, se reproduce el audio que avisa de un nuevo mensaje, como se ha explicado anteriormente, y se muestra dicho mensaje en una MessageBox. Al cerrar la MessageBox, se limpia el texto de la textbox y se vuelve a mostrar el formulario de Escritura.

El botón Salir cierra el formulario. Se ha intentado ejecutar un script que cierre el teclado en pantalla pero en las versiones más recientes de windows no se puede matar ese proceso sin la contraseña de superusuario del ordenador en el que se ejecuta el programa, por tanto se debe cerrar el teclado pinchando en la X del mismo.

7.5. ClickSpeak

ClickSpeak ha sido codificado en C++ en el entorno de programación de Visual Studio 2010. Para crear la interfaz gráfica se han utilizado formularios de Windows Form.

La aplicación consta de cinco formularios, los cuatro de EyeSpeaker más otro de configuración:

- Bienvenida
- Frases
- Respuestas
- Escritura
- Configuración

7.5.1. Bienvenida

Este formulario consta de siete etiquetas (label), un temporizador (timer) para poder mostrar las diferentes opciones, un contador (integer) para determinar la opción que se está mostrando para crear el nuevo formulario de la opción correspondiente, un objeto de tipo Configuración y un botón que permite acceder al panel de configuración.

Las dos primeras etiquetas son texto plano a modo informativo y nunca cambian. Las cuatro siguientes son las posibles opciones de comunicarse que permite la aplicación más la opción de salir. Están por separado para poder sombrear cada una de ellas según el valor del contador.

La última etiqueta muestra la opción de menú sombreada en cada paso correspondiente al valor del contador.

Cuando ocurre el evento Activated (formulario activo), se ejecuta el método InitializeTimer. En este método se inicializa el intervalo del temporizador al valor que indica el atributo del objeto Configuración, se activa poniendo Enabled a True y se invoca a la función timer1 que responde al evento de temporizador cumplido.

La función timer1 ejecuta un bloque switch según el valor del contador, llamado contadorLabel. Dependiendo del valor de este contador, el texto de la última etiqueta tendrá una cadena de texto u otra y se sombreará, mediante la propiedad BackColor de cada label, la opción correspondiente del menú; después de ejecutar el case correspondiente se incrementa el valor del contador y se controla que sea una valor entre 0 y 3 aplicándole la operación módulo 4.

La función InitializeTimer se ejecuta indefinidamente hasta que ocurre el evento KeyPress (pulsar una tecla) o MouseDown (click de ratón). Cuando esto ocurre, se detiene el temporizador, se preparan los formularios y, dependiendo del valor del contadorLabel, se muestra la opción correspondiente tras "esconder" la pantalla de Bienvenida. En caso de elegir la opción Salir, se cierra la aplicación.

Cuando se cierre el formulario de la opción elegida, se vuelve a mostrar la pantalla de bienvenida, se reinicializa el contador y se invoca a la función InitializeTimer para poder elegir de nuevo una opción.

7.5.2. Frases

Al igual que el formulario anterior, consta de siete etiquetas, un temporizador, un valor para el intervalo del temporizador y un contador.

También se utiliza la función InitializeTimer exactamente de la misma manera cuando ocurre el evento Activated. Esto se ejecuta indefinidamente hasta que ocurre el evento KeyPress o MouseDown, momento en el que se ejecuta un bloque switch en el que cada case invoca al método PlaySound correspondiente, la opción Atrás cierra el formulario con el método close.

La función InitializeTimer, además de sombrear cada opción del menú, mostrará en el formulario una imagen correspondiente a cada frase.

7.5.3. Respuestas

Su funcionamiento es análogo al del formulario Frases. Añadiendo el cambio del color del fondo del formulario mediante la propiedad BackColor.

7.5.4. Escritura

Este formulario consta de tres etiquetas, un temporizador, un valor para el intervalo y un contador; aunque la tercera etiqueta no se ve al cargar el formulario, ya que es la frase que "escribe" el usuario.

Se usa la función InitializeTimer descrita anteriormente y se invoca a la función Timer1. En este caso la función Timer1 utiliza una variable local, un array de string en el que se almacenan las letras del alfabeto en cada posición del mismo. Mediante sentencias if else if se cambia en la segunda etiqueta el texto de la misma, usando el contador como índice del array de letras; además se permiten las opciones de Espacio, Borrar y Enviar, controlando el incremento del contador con una operación módulo 30.

Para poder mostrar los caracteres en la etiqueta, es necesario crear un dato de tipo String^ a partir de los valores string del array.

Cada vez que se pulse una tecla o botón del ratón, evento KeyPress o MouseDown, se escribirá un carácter en la última etiqueta, incluyendo el espacio, dependiendo del valor del contador cuando se pulsó la tecla; se borrará el último carácter invocando a la función Remove de String o se mostrará el mensaje en un MessageBox y se reproducirá el audio de aviso mediante la función PlaySound.

Para añadir el texto se utiliza una variable local, un array de string en el que se almacenan las letras del alfabeto en cada posición del mismo, además del carácter espacio; para poder escribirlo en la etiqueta es necesario pasar de string a String^ igual que se hace al mostrar las letras en la segunda etiqueta.

Tras mostrar el mensaje y avisar con el audio, se cierra el formulario y se muestra de nuevo la pantalla de Bienvenida.

7.5.5. Configuración

El formulario de Configuración consta de tres botones y un integer. Estos botones establecen un valor para el integer que almacena el intervalo de tiempo que utilizan los temporizadores de los demás formularios. Tras notificar el cambio en un MessageBox, se cierra el formulario para volver a la pantalla de Bienvenida.

El programa ClickSpeak ha sido probado con éxito, funcionando especialmente bien la elección Si/No y la selección de estados de ánimo, aunque es insuficiente para completar sus necesidades expresivas, especialmente los sentimientos o la elección de tema de su interés, que sigue siendo muy complicado de averiguar, como elección de películas, canales de TV, libros para leerle, si quiere estar tranquilo.

Moisés Mateos es un niño de 12 años con diagnóstico de parálisis cerebral infantil severa por sufrimiento perinatal (PCI), nació en parada cardiorespiratoria estimada en 20 minutos que le provocó una hemorragia cerebral, que le afectó, sobretodo, al habla y la deglución, por lo que es alimentado por gastrostomía, teniendo una sonda en el estomago.

El programa de eyetraquing no ha podido probarlo por qué requiere tener la cabeza quieta, y Moisés padece de atonía cervical y estrabismo, por lo que solo puede hacer ciertos movimientos fluctuantes, además mira de lado.

Hasta ahora se comunican con él con asentimientos de la cabeza y toques con las manos, pudiendo elegir pictogramas con hasta 5 opciones. También asiente con la risa. Cuando su disponía lo permite, gracias a un baño de agua fría puede articular palabras como si, no, mamá, hambre, agua o frío. No tiene afectadas las capacidades cognitivas y conoce las letras y los números, incluso aprendió a sumar. También tiene sentido del humor y sabe mentir. Distingue el día de la noche y conoce el tiempo atmosférico.

Cuando no puede expresar un sentimiento o elección se frustra, se le activa el reflejo de Moro que suele perderse al nacer, y se auto lesiona hiriéndose en la cara. Dicho reflejo se le activa con el calor, la oscuridad, el ruido y la mencionada frustración por la incapacidad de comunicarse.



Ilustración 18 - Moisés probando ClickSpeak

Para este proyecto no se ha aplicado una metodología concreta de desarrollo de software aunque la más semejante es la que se conoce como ExtremeProgramming. Esta metodología se basa en la codificación del código a la vez que se van estableciendo los requisitos del sistema, aceptando el continuo cambio de los requisitos y la modificación y expansión del código durante todo el proceso de desarrollo.

En un primer momento se realizó una reunión entre el director y la codirectora del proyecto para establecer los objetivos del trabajo. En dicha reunión se decidió crear un software de seguimiento de pupila para que personas con parálisis cerebral pudieran comunicarse a través del ordenador u otro dispositivo.

Una vez establecido el punto de partida comenzó la investigación sobre la tecnología de EyeTracking, todo ello está detallado en el apartado de Estado del arte, en la introducción de esta memoria. También se investigó sobre las aplicaciones software que ya existían hasta el momento, como se ha descrito en el apartado de Programas probados, y se optó por crear una interfaz que cumpliera el objetivo del trabajo utilizando el ratón de mirada que proporciona el software HeadMouse.

Una vez que se desarrolló la aplicación EyeSpeaker, nos dimos cuenta de que la vista no era el mejor método de interacción para las personas a las que iba destinada el proyecto así que se concertó una nueva reunión para establecer cómo se debería construir el nuevo software. En dicha reunión se decidió que los sistemas de interacción entre el usuario y la aplicación serían clicks de ratón y pulsaciones de teclado.

Primero se desarrolló el programa KeySpeaker que funcionaba con pulsaciones de teclado, y a continuación, como programa independiente, MouseSpeaker, que ejecutaba la misma funcionalidad pero con clicks de ratón. Una vez que se comprobó que ambos programas funcionaban correctamente, se fusionaron ambos métodos de interacción dando lugar al programa definitivo, denominado ClickSpeak.

Finalmente, Moisés realizó pruebas con ClickSpeak para comprobar que el sistema era adecuado para sus intereses. Como se ha mencionado en el apartado anterior, estas pruebas fueron satisfactorias aunque un poco insuficientes ya que el programa no permite demasiadas opciones en cuanto a la expresión de frases predeterminadas o sentimientos, y la opción de Escritura parece ser demasiado pesada, lenta y un poco incómoda.

La memoria ha sido redactada y modificada cuando ha sido necesario, durante todo el desarrollo del proyecto. Empezando por el Estado del arte antes de programar, especificando el uso y funcionamiento de los sistemas durante la codificación, y completando el resto de apartados después de que se ha comprobado el correcto funcionamiento de los programas.

10. TRABAJOS FUTUROS

Para el curso 2016/17, un grupo de tres alumnos continuará con este proyecto como Trabajo de Fin de Grado. Su cometido será desarrollar la aplicación con el lenguaje JAVA y tecnologías Web, además de hacerlo compatible con dispositivos móviles como tablets; además contará con archivos XML para configurar todos los aspectos de la aplicación.

11. CONCLUSIONES

La idea original de este trabajo, como indicaba su título original, era construir un software de seguimiento de pupila; pero finalmente se descartó esta idea por dos motivos principales. El primero es la complejidad de construir un software de estas características y sin experiencia previa en este campo, la segunda es que este tipo de software no sería el más indicado para la persona que lo iba a utilizar debido a la enfermedad que padece.

Por ello se adoptó la decisión de modificar el método de interacción entre el usuario y el sistema. Como se ha mencionado anteriormente, estos métodos son pulsaciones de teclado y clicks de ratón.

Concretamente, los clics de ratón ha resultado ser el mejor método ya que los niños que padecen este tipo de enfermedades, a menudo utilizan unos pulsadores como medio para comunicarse, véase el que aparece en la siguiente foto.



Ilustración 19 - Pulsador

Estos pulsadores tienen un cable jack que se puede acoplar a algunos ratones específicos, de esta manera se pueden realizar los clicks del ratón accionando el pulsador, lo que facilitará enormemente la interacción entre el usuario y el sistema que se ha desarrollado en este trabajo.



Ilustración 20 - Pulsador conectado a ratón

Tras realizar las pruebas con el niño al que iba destinado el programa, podemos decir que se ha desarrollado una herramienta útil pero algo pobre en cuanto al objetivo de permitir a esta persona que pueda comunicarse de forma completa con el resto de personas. Es por ello que este Trabajo de Fin de Grado se puede considerar un primer paso, o prototipo, de un futuro sistema de comunicación, cuya licencia es de código abierto y se puede descargar desde SourceForge.

11.1. Conclusions

As indicated in the original title, the first idea for this project was to build a Eye Tracking software. The complexity of a software needed, without experience on the field, and the poor adaptation of the program to the needs of the subject's disease were the two main reasons to modify the connection metods between user and system. As it has been said before, the communication will be done by key pressing and mouse clicks.

Specifically the mouse have shown to be best for adaptation, given the familiarity that children with this kind of disabilities already have with buttons for their communication, like the one of the following picture.



This kind of buttons are connected by a jack wire, that can be added to some specific mouse, so it will act as the pulsator using the mouse clicks, accomplishing the same functions, and improving the interaction between the user and the system of this project enormously.



After the child subject took the tests, we are able to conclude that a useful tool has been developed, although it is not yet complete for providing people with this kind of disease the ability to communicate easily with other people. Thats why this Grade Project can be considered a first stage prototype of a future communication system with open source license. This proyect is avalaible to downdload on SourceForge.

- [1]. Seguimiento de ojos
 - https://es.wikipedia.org/wiki/Seguimiento_de_ojos
- [2]. Mapas de calor y zonas ciegas
 - <http://www.usolab.com/articulos/eyetracking-usabilidad-comunicacion.php>
- [3]. Neuromarketing
 - <http://neuromarca.com/neuromarketing/eye-tracking/>
- [4]. Tobii PCEye
 - <http://www.tecnoaccesible.net/content/tobii-pceye>
- [5]. TrackEye
 - <http://www.codeproject.com/Articles/26897/TrackEye-Real-Time-Tracking-Of-Human-Eyes-Using-a>
- [6]. FaceTrackNoIR
 - <http://facetracknoir.sourceforge.net/home/default.htm>
- [7]. QMouse
 - <http://www.ecured.cu/index.php/QMouse>
- [8]. HeadMouse
 - <http://robotica.udl.cat/headmouse.html>
- [9]. Windows Form, C# y C++
 - <https://msdn.microsoft.com/>