

Aplicación Web de secuenciador musical basado en datos de los movimientos oculares (*eye-tracking*)

Jonathan Manuel Fraire-López¹, Francisco López-Orozco^{1,2},
Jesús Israel Hernández-Hernández^{1,2}, Juan Gabriel Ruiz-Ruiz³

¹ Universidad Autónoma de Ciudad Juárez,
División Multidisciplinaria en Ciudad Universitaria,
México

² Universidad Autónoma de Ciudad Juárez,
Maestría en Cómputo Aplicado,
Instituto de Ingeniería y Tecnología,
México

³ Universidad de la Sierra Juárez,
México

{al159943}@alumnos.uacj.mx
{francisco.orozco,israel.hernandez}@uacj.mx
{jugaruiz}@unsij.edu.mx

Resumen Este artículo presenta el desarrollo de una aplicación web de un secuenciador de instrumentos musicales que se utiliza con datos del movimiento de los ojos. Una ventaja de la aplicación es la habilidad para reproducir y grabar los instrumentos usando una webcam convencional. El proyecto está basado en *WebGazer* una biblioteca de software libre escrita en javascript. A partir de ésta se obtienen los datos de entrada de la vista por parte de los usuarios y se estima la posición de la mirada sobre la pantalla de la computadora. Las fases de prueba y validación fueron llevadas a cabo con 30 músicos experimentados. Los resultados muestran que la propuesta es correcta y aceptada por estos usuarios. Sin embargo, nuestro desarrollo puede ser mejorado en varios aspectos como la inclusión de más instrumentos musicales entre otros.

Palabras clave: Interacción basada en la vista, eye tracking, secuenciación de instrumentos musicales, interacción por cámara web.

Web Application of a Music Sequencer based on Eye Movement Data (Eye-Tracking)

Abstract. This paper presents the development of a web-based application for a musical instrument sequencer controlled through eye movement data. A key advantage of the application is its ability to play and record instruments using a standard webcam. The project is based

on WebGazer, an open-source JavaScript library. Using this library, gaze data from users is captured, and the system estimates the position of the user's gaze on the computer screen. Testing and validation phases were conducted with 30 experienced musicians. The results show that the proposed system is functional and well received by users. However, the application can be improved in several aspects, such as the inclusion of more musical instruments, among others.

Keywords: Gaze-based interaction, eye tracking, musical instrument sequencing, webcam interaction.

1. Introducción

La música ha evolucionado a través del tiempo así como los instrumentos que se utilizan para su ejecución. Algunas herramientas de software usadas en esta área han también evolucionado y se han convertido de mucha importancia para los músicos hoy en día. Todo esto ha sido ocasionado entre otras razones por el incremento de la digitalización musical en nuestros días. De la misma forma, algunas tecnologías también han cobrado importancia como las del seguimiento ocular o *eye-tracking* [11]. El *eye-tracking* ha sido previamente usada en el ámbito musical aunque no ha alcanzado un suficiente grado de popularidad entre los músicos todavía. En este proyecto se propone una solución a uno de los problemas o dificultades actuales del uso de la tecnología *eye-tracking* en la música. El *eye-tracking* ha fallado en su intento por captar la atención de usuarios dentro del campo musical. Esta dificultad refiere al hecho de que las aplicaciones existentes para reproducir o registrar instrumentos, requieren el uso de un eye-tracker especializado, al que usualmente no tienen acceso la mayoría de los usuarios [12]. Por esta razón, este desarrollo es una propuesta de software para la reproducción de música a partir de una cámara web convencional sin la necesidad de un eye-tracker.

2. Trabajos previos

La interpretación de la música es normalmente asociada a las extremidades del cuerpo o la voz, pero hoy en día hay diferentes alternativas a estas extremidades. Por ejemplo, hay instrumentos musicales virtuales que pueden tocarse sin la necesidad de las extremidades, usando sensores o la vista [1]. De esta forma se ofrecen nuevas alternativas a los músicos, teniendo la oportunidad de contar con uno o más elementos para desarrollar cierta independencia musical. Una ventaja podría ser usar la vista para tocar un instrumento y una extremidad para tocar otro al mismo tiempo. Adicionalmente, esto abre la posibilidad para que personas con capacidades diferentes y a las que se les imposibilita tocar un instrumento musical físico, puedan aventurarse en el campo de la música. La necesidad de adaptar y simplificar la forma en la que los instrumentos musicales virtuales son tocados o ejecutados ha permitido la experimentación

con secuenciadores virtuales, para grabar diferentes pistas que se reproducen repetidamente y así con más posibilidades de reproducir otros instrumentos sobre las pistas cuando se están ejecutando [2].

Diferentes trabajos y proyectos han sido desarrollados en este campo haciendo énfasis en instrumentos musicales y secuenciadores en la Web. Un secuenciador con características interesantes es *Soundation* [3], una herramienta de pago dedicada a la producción musical y edición de pistas. Los instrumentos son ejecutados solo con controladores físicos como con *Musical Instrument Digital Interfaces (midi)*. Un secuenciador muy práctico es Online Sequencer [4], cuya interfaz basada en una cuadrícula que segmenta los ritmos en el cual las notas se dibujan en tiempos específicos para todos los compases. Otro ejemplo es *midi.city* [8], una herramienta gratuita que permite secuenciar un número de instrumentos musicales. También encontramos *EyeHarp* [5], una aplicación de escritorio no gratuita que permite la ejecución de instrumentos con la vista. Requiere de un *eye tracker* para ejecutar los instrumentos y navegar dentro de la aplicación con la vista.

Por otro lado, se pueden encontrar dos aplicaciones de Microsoft para la secuenciación musical: *Microsoft Hands-Free Sound Machine* [6] y *Microsoft Hands-Free Sound Jam* [10]. Ambas son aplicaciones gratuitas que requieren un *eye tracker* para funcionar. El primero es un secuenciador que combina sonidos multimedia previamente creados con instrumentos musicales de la aplicación y el segundo está más orientado a la composición musical.

En el proyecto que se describe en el presente artículo, el objetivo fue desarrollar una alternativa de uso del *eye tracking*. La propuesta es una aplicación web para la ejecución y secuenciación de instrumentos musicales mediante la vista usando una cámara web convencional. Se plantearon dos preguntas importantes: ¿cuál sería el rendimiento de la aplicación web y cuáles serían las ventajas de usar este secuenciador comparado con algún controlador para un instrumento musical?

3. Metodología e implementación

La metodología de cascada fue usada para el desarrollo de este proyecto con las siguientes etapas: análisis de requisitos, diseño, codificación, pruebas y mantenimiento. Por las características de este modelo secuencial, cada etapa comienza y termina antes de empezar la siguiente [7].

En la primera etapa, se llevó a cabo la elicitación de requerimientos de la aplicación, definiendo los funcionales como la *Reproducción y Grabación de instrumentos* y no funcionales como la *Calibración de la biblioteca* o la *Webcam necesaria*, además de la elaboración de diagramas de casos de uso. Los casos de uso describen la manera en la que el usuario puede interactuar con el sistema y de las funciones disponibles para él.

Durante la etapa de diseño se recolectaron las muestras o *samples*, se diseñaron los instrumentos, el menú y el diseño general de la aplicación.

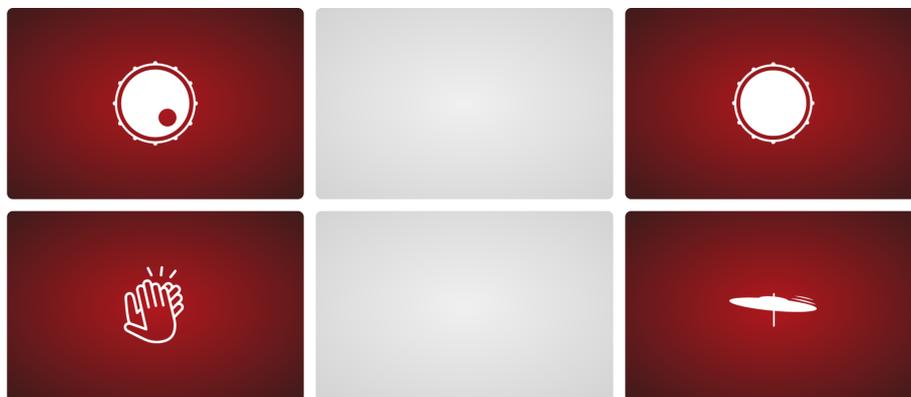


Fig. 1. Diseño de instrumentos.



Fig. 2. Controles del secuenciador.

Cada nota o elemento de un instrumento musical corresponde a un *sample* en un archivo de audio MP3. Para el caso de los instrumentos piano, guitarra y bajo se utilizó la aplicación Logic Pro X, la cual cuenta con una gran variedad de instrumentos virtuales que pueden ser ejecutados vía MIDI. En el caso de la batería se usaron *samples* gratis obtenidos de la página de SamplesWrap [9]. Una vez recolectadas las muestras fue necesario implementar una interfaz de cada instrumento en forma de *pad* rectangular. En la Figura 1 se muestran los cuatro *pads* a los costados y en el centro dos *pads* de descanso donde los usuarios pueden mirar sin que la aplicación lance sonido alguno. Esto último fue basado en las sugerencias aportadas en [14]. Posteriormente se diseñó un menú estático de forma horizontal cuyo apariencia y botones pueden apreciarse en la Figura 2.

Finalmente en la Figura 3 se presenta la interfaz principal del secuenciador dividido en 3 partes. En la sección 1 aparece el visualizador de la cámara web del usuario en tiempo real de la biblioteca *WebGazer*. En 2, corresponde al menú de opciones y en la 3 aparece el instrumento seleccionado. La codificación del secuenciador fue dividido en dos partes principales: el secuenciador en sí y por otro lado la parte de la biblioteca *WebGazer* que se utilizó para la ejecución con la vista. Para el desarrollo del secuenciador se usaron diferentes bibliotecas. En la Figura 4 se muestra el orden del flujo de datos y conexión de las bibliotecas utilizadas para la función del secuenciador. La primera biblioteca para la programación del secuenciador fue *Howler JS*, usada para cargar los *samples*. Al seleccionar cada nota o elemento del instrumento se lanza el sonido de éste. Para evitar retrasos y lograr que la interpretación sea en tiempo real

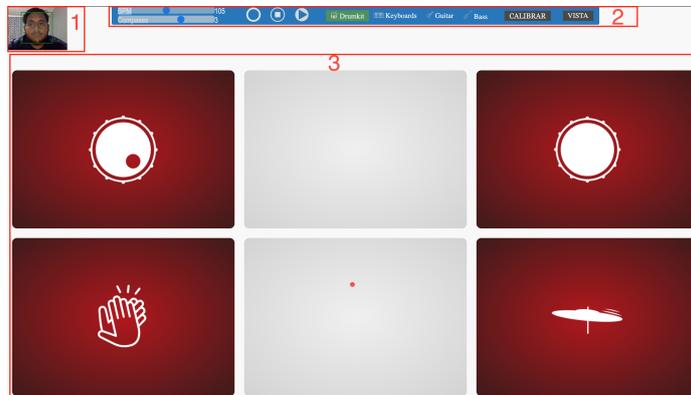


Fig. 3. Interfaz general de la aplicación.

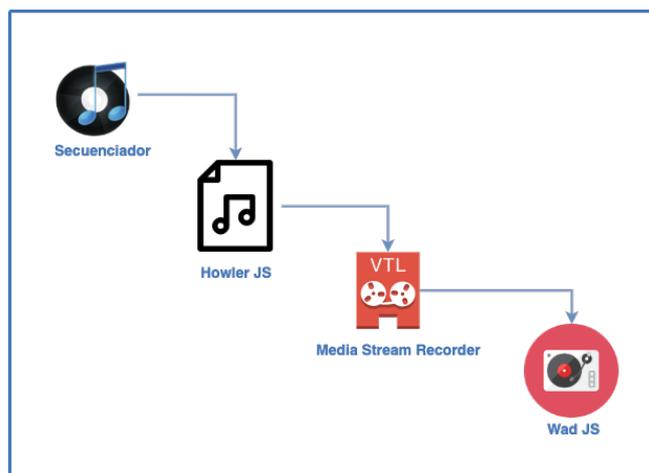


Fig. 4. Bibliotecas utilizadas.

se usó *Howler JS*, la cual provee la función de crear *sprites*. Estos consisten en juntar y unir cada archivo de audio MP3 o un archivo WebM en uno solo y así crear un archivo tipo JSON, donde reconoce el segmento del archivo correspondiente y lo separa en segundos, para así llamar siempre al mismo archivo desde diferentes partes.

Una vez definidos los *sprites* de cada instrumento del secuenciador, se pasa la información de la biblioteca *Howler JS* a la biblioteca *Media Stream Recorder*. Esta es la encargada de grabar el audio. Básicamente esta biblioteca y *Howler JS* funcionan mediante entrada y salida. Se tomó un objeto de salida de la biblioteca de *Howler JS* y se pasó a la entrada de *Media Stream Recorder*. Una vez con los



Fig. 5. Pantalla de inicio de la aplicación.

datos dentro de esta biblioteca, estos están listos para ser procesados y enviados a la salida. Asimismo, la salida de *Media Stream Recorder* se conectó a la entrada de la biblioteca *Wad JS*. Esto fue posible porque se creó un objeto tipo *Wad*, el cual recibe como fuente de entrada el objeto tipo *blob* que genera la salida de *Media Stream Recorder*.

Para iniciar la grabación se llama a la función `.start()` de la biblioteca *Media Stream Recorder*. Después de la grabación que el usuario hace, se lanza un evento de la biblioteca *Media Stream Recorder*, y justo cuando la grabación se termina se llama a este evento llamado `.onStop()`, en el cual se genera el archivo tipo *blob* que se menciona anteriormente. Dentro de este mismo evento cada grabación se va guardando en un arreglo donde cada posición contiene un archivo tipo *Wad* que representa la grabación hecha.

Después de la programación de las funciones del secuenciador, se realizó la conexión con la biblioteca *WebGazer*. Esta biblioteca es la encargada de estimar la posición de dónde el usuario está mirando en la pantalla y permite conectarla con el secuenciador, de tal forma que los sonidos puedan ejecutarse al momento que el usuario fije la vista en un elemento o nota de un instrumento. Cabe hacer mención que la etapa de pruebas y evaluación se apoyó de la aplicación de un *After-Scenario Questionnaire (ASQ)*, que se detalla más adelante en la sección 4.

3.1. Secuencia de uso de la aplicación

Al acceder a la aplicación, lo primero que se encuentra es la pantalla principal de inicio como se aprecia en la Figura 5. La pantalla principal cuenta con el menú de opciones, dentro del cual se encuentra el botón **CALIBRAR** que se encarga de dirigir a la pantalla del sistema de calibración para la biblioteca *WebGazer*. Una vez que se accede a la pantalla de calibración, es necesario completarla para poder regresar a la pantalla principal. El proceso de calibración consta de seguir con la mirada el cursor y dar 5 clics en cada uno de los puntos grises y así estos

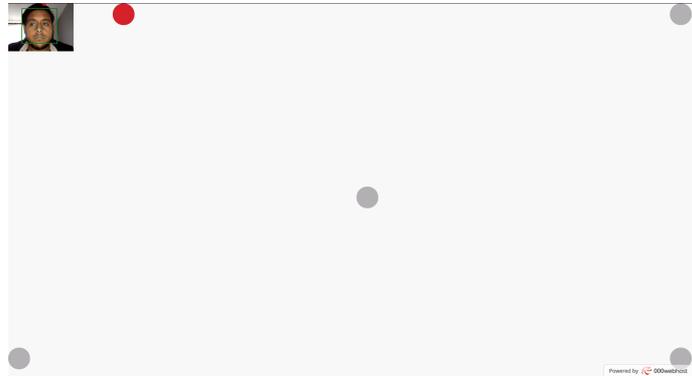


Fig. 6. Pantalla de calibración.

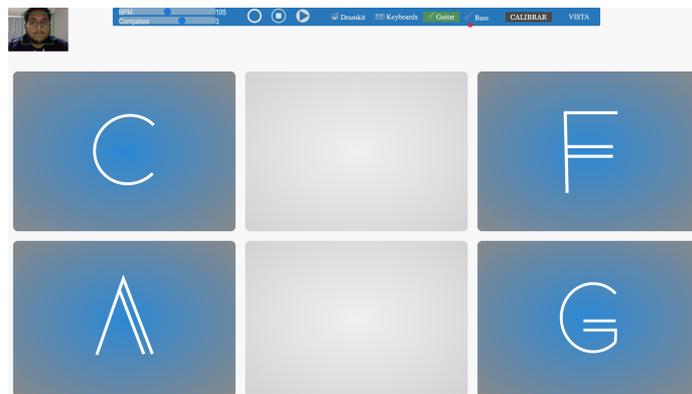


Fig. 7. Pantalla de instrumento seleccionado.

cambiarán a color rojo como se muestra en la Figura 6. Terminada la calibración, la aplicación regresará a la pantalla de inicio como se puede ver en la Figura 7, donde se puede seleccionar cualquier instrumento para ser ejecutado o grabado.

4. Resultados y discusión

En esta sección se presentan los resultados de las pruebas realizadas con 30 músicos que pudieron manifestar su aceptación o rechazo de la aplicación web. De estos, 28 fueron hombres y 2 mujeres entre 17 a 38 años. Todos, experimentados en al menos uno de los cuatro instrumentos que tiene la aplicación: bajo, batería, teclado y guitarra. A cada uno de los músicos se les pidió realizar dos tareas y la evaluación se llevó a cabo tanto de forma cualitativa como cuantitativa. En el

Tabla 1. Promedio de respuestas en evaluación cualitativa para Tarea 1 y Tarea 2.

Pregunta	Tarea 1 (\bar{x} σ)	Tarea 2 (\bar{x} σ)
1	5.8 1.09	5.36 1.27
2	5.73 1.14	5.63 1.06
3	5.76 1.30	5.66 1.09

primer caso, mediante un cuestionario tipo *After-Scenario Questionnaire (ASQ)* que se aplicó después de cada tarea completada, para que cada usuario reflejara su grado de satisfacción con la aplicación web. La evaluación cuantitativa fue basada en dos métricas: porcentaje de tareas completadas y tiempo que le llevó realizar cada tarea. La aplicación desarrollada es accesible a partir de un *host* gratuito del proveedor *000WebHost*[13] desde la siguiente URL 1. El código fuente del proyecto es accesible en URL 2.

4.1. Tarea 1

En esta tarea se les pidió a los usuarios que grabaran cualquier progresión de acordes a su gusto, mediante clics del ratón. La primera pregunta para los usuarios después de la tarea fue: **En general, ¿Qué tan satisfecho estoy con la facilidad de completar la tarea en este escenario?** La escala de respuestas para esta pregunta fue de 1 (totalmente en desacuerdo) hasta 7 (totalmente de acuerdo). El promedio de las respuestas fue 5.8 ($\sigma = 1.09$), que se inclina hacia el contexto totalmente de acuerdo.

Las respuestas más frecuentes fueron 7, 6 y 5, con 10, 8 y 9 usuarios respectivamente, lo que significa que la mayoría de usuarios están conformes con la facilidad de usar la aplicación al realizar la tarea. La segunda pregunta fue: **En general, ¿Qué tan satisfecho estoy con la cantidad de tiempo que tomó completar la tarea en este escenario?** En este caso, nuevamente los resultados reflejan que la mayoría de usuarios están satisfechos con el tiempo que les tomó realizar la tarea, dado que las respuestas fueron 7, 6 y 5, con 10, 7 y 9 usuarios respectivamente. El promedio de respuestas de los usuarios fue 5.73 ($\sigma = 1.14$), con una tendencia hacia el lado totalmente de acuerdo. Para el caso de la pregunta tres, el promedio fue 5.76 ($\sigma = 1.30$), la cual cuestiona: **¿Qué tan satisfecho estoy con la información de soporte (ayuda en línea, mensajes) al completar la tarea?** La respuesta más frecuente fue el número 7 con 11 votos, seguida del número 6 con 8 y el número 5 con 7 votos de usuarios.

4.2. Tarea 2

En la segunda tarea se graban los instrumentos con la vista. Sin embargo, al final el mismo cuestionario fue aplicado a los usuarios. Acerca de la primera pregunta de la facilidad para completar la tarea, el promedio de respuestas fue de 5.36 ($\sigma = 1.27$), el cual a diferencia de la Tarea 1 (5.8, $\sigma = 1.09$), fue menor,

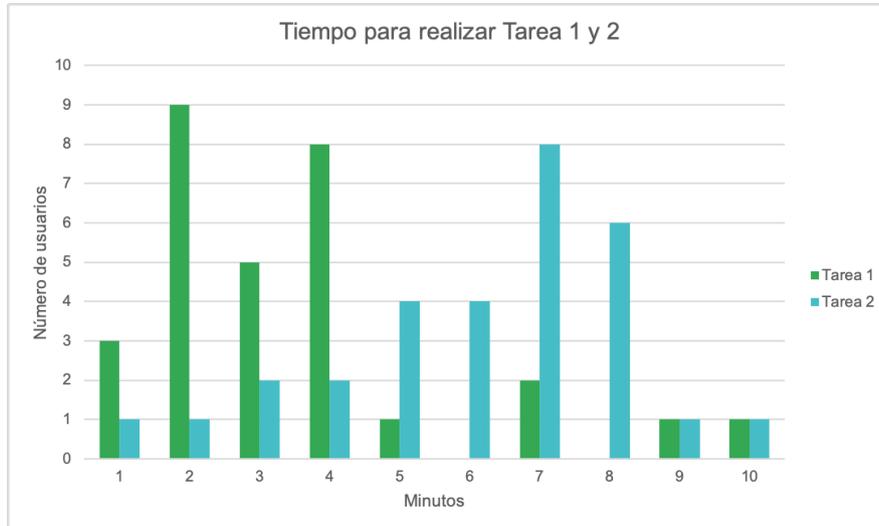


Fig. 8. Comparativa de tiempo para realizar ambas tareas.

pero se mantiene inclinado hacia el contexto totalmente de acuerdo, mostrando como resultado más frecuente el número 6. Para la segunda pregunta se tiene también como respuesta más frecuente el número 6 al igual que en la pregunta anterior, además de un promedio de 5.63 ($\sigma = 1.06$). Para la última pregunta, nuevamente el resultado más frecuente es 6, con promedio 5.66 ($\sigma = 1.09$), lo cual, al igual que en las otras 2 preguntas, es un resultado un poco más bajo a comparación con la Tarea 1, lo cual refleja menos satisfacción para realizar la tarea pero no llegando a ser un mal resultado sobrepasando la mitad de la escala, encontrándose inclinado hacia el contexto totalmente de acuerdo. La Tabla 1 resume estos resultados.

Acerca de la evaluación cuantitativa, como ya se había mencionado anteriormente, los resultados se distribuyeron a través de 2 variables: el porcentaje de tareas completadas por el usuario y el tiempo que tomó el usuario para realizar la tarea. Ambas tareas se completaron al 100% por los 30 usuarios. Respecto a la segunda variable, en la Figura 8 se observa que la mayoría de usuarios tardaron alrededor de 2 y 4 minutos en realizar la Tarea 1, con un tiempo promedio de 3.53 min ($\sigma = 2.20$). La segunda tarea les llevo más tiempo a los usuarios en realizarla, en promedio 6.13 min ($\sigma = 2.09$). En la Figura 8 se observa que les tomó entre 7 y 8 minutos. Además, se puede observar una comparativa de ambas tareas, reflejando el número de usuarios y el tiempo que les tomó realizar cada tarea.

La mayoría de usuarios tardaron poco tiempo en realizar la Tarea 1 y que pocos tardaron mucho tiempo en completarla. En la Tarea 2 se presenta lo contrario, muchos usuarios tardaron mucho tiempo y pocos usuarios pudieron

realizarla en un tiempo corto. Este comportamiento se pudo deber al grado de familiarización de los músicos con aplicaciones relacionadas con el seguimiento ocular, la mayoría de usuarios presentaban un grado bajo o nulo de experiencia con la función de ejecutar los instrumentos con la vista la cual era el diferenciador de la Tarea 2 con la Tarea 1. Debido a esto, al no haber usado antes una aplicación como la que describe este proyecto, los usuarios tendían a tomar más tiempo para familiarizarse primero con la función además de corregir factores externos como puede ser la iluminación del entorno para que la biblioteca *WebGazer* pudiera captar sin problema el rostro, postura del cuerpo y rostro frente a la *webcam*, distracciones que daban como resultado el perder la calibración, entre otros. Una vez que ya se encontraban familiarizados procedían a realizar la grabación.

5. Conclusiones

En el presente artículo se presentó el desarrolló un secuenciador musical web, compuesto de cuatro instrumentos musicales que pueden ser ejecutados y grabados mediante la vista a través de una *webcam* con la biblioteca *WebGazer* que se encarga de estimar el seguimiento ocular del usuario. En este desarrollo no se requiere de un dispositivo *eye-tracker* como en el caso de trabajos reportados en el área y citados en este artículo. El objetivo de este proyecto fue desarrollar una aplicación web para secuenciar instrumentos musicales mediante la vista a través de una *webcam* convencional.

Además se plantearon los siguientes objetivos específicos:

- diseñar un secuenciador web con los instrumentos musicales convencionales,
- determinar el tamaño y separación adecuada de los instrumentos musicales virtuales y
- establecer el protocolo de comunicación entre los instrumentos musicales y la *webcam* para el manejo del secuenciador.

Todos fueron cumplidos gracias a la biblioteca *WebGazer*. Después de la fase de evaluación del proyecto con los usuarios profesionales de la música, se concluye que el desempeño del secuenciador es bueno, dado que la mayoría de los usuarios manifestaron estar totalmente de acuerdo con la aplicación.

También se puede destacar que la desventaja principal es que los usuarios no están familiarizados con la función para ejecutar los instrumentos con la vista, ya que existen controladores para ejecución de instrumentos más comunes, además de otras desventajas, como lo son factores externos que se mencionan y se describen anteriormente en este artículo.

Sin embargo, a pesar de que el proyecto final es un producto totalmente funcional, aún existen mejoras que se pueden hacer para enriquecer futuros proyectos o productos similares. Algunas de estas mejoras podrían ser el desarrollo de funciones extras de edición del audio grabado por el usuario, la creación de instrucciones o tutoriales más ilustrativos, que puedan captar la atención del usuario y sean de ayuda para el correcto uso de la aplicación, así como la adición de funciones MIDI para uso de controladores externos.

Referencias

1. Z. Vamvakousis, R. Ramirez: The EyeHarp: An eye-tracking-based musical instrument. In: Proceedings of the 8th Sound and Music Computing Conference, SMC 2011 (2011)
2. J. Palazón-Herrera: Creación musical en Educación Secundaria a través del uso desecuenciadores multipista (2016)
3. Soundation — Make music online. (2021) [Online]. Available: <https://soundation.com/>.
4. Make music online. (2021) [Online]. Available: <https://onlinesequencer.net/>.
5. Inicio - The EyeHarp. (2021) [Online]. Available: <http://theyeharp.org/es/inicio/>.
6. Microsoft Hands-Free Sound Machine. (2021) [Online]. Available: <https://www.microsoft.com/en-us/research/project/microsoft-hands-free-sound-machine/>.
7. W. Van Casteren: The Waterfall Model and the Agile Methodologies: A comparison by project characteristics. (2017)
8. midi.city - free online synth instrument. (2021) [Online]. Available: <https://midi.city/>.
9. Free audio samples, drum loops & kits, vocals, royalty free music. (2021) [Online]. Available: <https://sampleswap.org/>.
10. Microsoft Hands-Free Sound Jam - Microsoft Research. (2021) [Online]. Available: <https://www.microsoft.com/en-us/research/project/microsoft-hands-free-sound-jam/>.
11. A. Poole, L. Ball: Eye Tracking in Human-Computer Interaction and Usability Research : Current Status and Future Prospects (2004)
12. L. González and J. D. Velásquez, “Una aplicación de herramientas de eye-tracking para analizar las preferencias de contenido de los usuarios de sitios web, ”Revista de ingeniería e sistemas, vol. 26, no. 1, pp. 95–118 (2012)
13. Mejor Hosting Gratis de Mexico 2020 — Hosting Gratuito. (2020) [Online]. Available: <https://mex.000webhost.com/>.
14. C. A. Gainer, R. W. Obermayer: Pilot Eye Fixations While Flying Selected Maneuvers Using Two Instrument Panels. Human Factors, vol. 6, no. 5, pp. 485–501, SAGE Publications Inc. (1964) Available:<https://doi.org/10.1177/001872086400600507>.