



UNIVERSIDAD GERARDO BARRIOS

Facultad de Ciencia y Tecnología

Ingeniería en Sistemas y Redes informáticas

Asignatura:

Programación Computacional III - PROYECTO FINAL

Docente:

Ing. William Alexis Montes Giron

Integrantes

-William Josué Guandique Rivera – SMSS076423.

-(A1) William Alfredo Irula González – SMSS141823.

-Emerson Aldahir Portillo Segovia – SMSS141223.

-Briseily Yamileth Solórzano Hernández – SMSS030723.

-Javier Alexander Vargas Díaz – SMSS150823.

San Miguel, 24 de noviembre de 2024

Descripción del proyecto:

Nuestro proyecto se centra en la creación de una aplicación tipo videojuego, el cuál incentiva al público a aprender LESSA de una manera dinámica, como grupo consideramos la importancia de la inclusión, y una forma de aplicarla es conociendo el lenguaje de señas, problema que nuestra aplicación busca reducir. Esto se aplica en la forma que el juego se desarrolla, pues además de una “biblioteca de señas” para conocer LESSA, está pensado en que de una manera aleatoria se seleccione una letra (como objetivo inicial está pensado en las vocales) y el usuario debe hacer la seña respectiva de dicha letra; seguidamente, la aplicación evaluará si la seña es correcta o no. Luego de este proceso, se espera generar en el usuario el gusto por conocer el lenguaje de señas y ponerlo en práctica.

Avances del proyecto:

Nuestra aplicación web lleva un avance considerable en las diferentes páginas html que contiene, ya que cada una posee su contenido respectivo y en su mayoría están 100% funcional (excepto las 3 páginas que contienen los 3 niveles de nuestro juego). Las páginas y sus funcionalidades son:

Prioridad baja:

- Equipo (grupo.html): hemos tomado a bien agregar esta página para dar información sobre el equipo desarrollador del proyecto. También incluye información sobre la aplicación.

Prioridad media:

- Home (index.html): muestra información sobre LESSA y cuán importantes es; además, tiene la opción para ir a “Juego” y a “Equipo” (ambas se encuentran en el NAVBAR).

- Secciones de uso (jugar.html): contiene lo fundamental de nuestra aplicación, con las opciones de ver el abecedario en LESSA y la opción para ir al juego como tal.

Prioridad alta:

-Abecedario (letras.html): está habilitada para recorrer el abecedario en LESSA. Funciona mediante los botones “anterior” y “siguiente” que pasan o regresan letras para ver su respectiva seña. Además, tiene un formato tipo teclado que permite seleccionar una letra y mostrar su seña. Esta sección es especial para nosotros pues hemos tomado a bien ser parte y ejemplo de nuestra iniciativa, que es practicar LESSA. Las fotos que contiene esta sección las hemos tomado los 5 miembros del grupo (tomando en cuenta que es nuestra primera experiencia, por lo que las señas podrían contener pequeños errores).

-Juego (selecNivel.html): presenta los 3 niveles de juego (principiante – vocales, medio – abecedario, avanzado – expresiones), el objetivo es cumplir con el nivel principiante, que es jugar con las vocales. Las otras 2 secciones están presentes como propuestas de lo que sería nuestra idea a largo plazo.

-Nivel principiante (nivelPri.html): es la sección de juego, donde la aplicación de una manera aleatoria pedirá una vocal y ahí mismo estará el frame de la cámara para su lectura, mediante nuestra programación con Python, se evaluará la seña capturada y se mostrará el resultado de acierto/desacierto.

-Nivel intermedio (nivelMed.html): es la sección de juego para abecedario, solo cuenta con la estructura del nivel principiante y no se pretende darle funcionalidad, ya que requiere de más cálculos que por tiempo es complicado realizarlo, o en su defecto, recursos para entrenar una red neuronal, tema que se nos complica aún más por nuestro equipo de software como también el tiempo que requiere su implementación (algunas letras se representan con movimiento, lo que aumenta su dificultad de programación).

-Nivel avanzado (nivelAvan.html): es la sección de juego para gestos, solo cuenta con la estructura del nivel principiante y no se pretende darle funcionalidad, ya que requiere entrenar una red neuronal compleja y de mucho peso para los equipos de software con los que contamos, como también el tiempo que requiere su implementación (otra complejidad es que aquí se toman en cuenta movimientos y evaluar gestos del rostro).

-app.py: es el que se debe ejecutar para iniciar nuestra aplicación, contiene las rutas a todas las páginas html del proyecto y la gestión de activación del frame de la cámara.

-lectorMano.py: contiene la programación de la funcionalidad de la cámara, como su lectura, proceso de dibujar la mano, seguir esta con un cuadro (área que será evaluada), entre otras funcionalidades. Es imperativo no ejecutar este archivo como principal, ya que no es el encargado del funcionamiento de la aplicación en general.

Adición de nuevo archivo: hemos optado por sustituir la idea del modelo entrenado de una red neuronal por **geometría computacional y trigonometría 3D**, pues la manera en que estamos detectando las señales es por medio de la posición de puntos de interés de la mano, que se pueden evaluar gracias a la herramienta Mediapipe, que cuenta con índices específicos para cada dedo y sus puntos. En este caso, el archivo que realiza todo este trabajo de análisis es el siguiente:

-detectorVocal.py: es el archivo más importante de este proyecto, ya que es el que realiza la lectura de la seña que capta la mano y compara el resultado con la letra que anteriormente este mismo código había solicitado.

Cabe mencionar que este archivo agregado, es muy completo, ya que indica al usuario el momento de iniciar y cómo inicia; además, al tocar enter obtiene el resultado y evalúa el acierto/desacierto.

Objetivos faltantes:

Prioridad alta:

- Combinar la funcionalidad que tiene el archivo 'detectorVocal.py' con el resto del proyecto; es decir, sustituir el archivo de uso actual (lectorMano.py) por el nuevo (detectorVocal.py). Este segundo se intentó implementar, pero tuvimos errores difíciles de solucionar. Una vez sustituido este archivo, el juego se desarrollará como está planificado.

Prioridad baja:

- Temas de diseño.

Plan de desarrollo:

Prioridad alta:

- Buscar la manera de implementar el archivo 'detectorVocal.py' en el proyecto.

Prioridad baja:

- Mejoras en el diseño de las páginas (parte visual).

Porcentaje de desarrollo: 90%.

Librerías necesarias: OpenCV, MediaPipe, Flask.