

TEXTBF COMPUTACIÓN UBICUA E INTELIGENCIA AMBIENTAL
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Proyecto de Realidad Aumentada



**UNIVERSIDAD
DE GRANADA**

Jose Luis Molina Aguilar

5 de julio de 2023

Curso 2022 / 2023
Correo: joselu201@correo.ugr.es

Índice

1	Descripción	3
2	Tecnologías	3
3	Requisitos	3
4	Mejoras para la colaboración entre jugadores	3
4.1	Colaboración y comunicación	4
4.2	Competencia estratégica	4
5	Puntos de la implantación	4
6	Implementación	5
7	TODO	10
8	Consciencia de Contexto	11
9	Ampliación Reconocimiento de Voz	12

1. Descripción

El objetivo de este proyecto es crear un juego de realidad aumentada en el que se utilizan marcadores ArUco como checkpoints similares a los del juego Mario Kart. El objetivo del juego es pasar por los marcadores en un orden aleatorio preestablecido en el menor tiempo posible. El juego pondrá a prueba la destreza al volante y la capacidad de memorizar y resolver problemas.

El orden de los checkpoints se generará aleatoriamente y se mostrará al principio del juego, pero no se volverá a mostrar. Deberás recordar el orden para completar el juego. Si crees que has sido capaz de memorizarlos en menos tiempo que el juego tiene definido, podrás pasar esa pantalla y comenzar. La gracia está en la capacidad y agilidad mental del jugador. El juego se iniciará cuando se detecte la palabra *Start*

2. Tecnologías

El proyecto utiliza OpenCV para la detección de marcadores ArUco y para el procesamiento de imágenes. También utiliza la biblioteca `face_recognition` para el reconocimiento facial, permitiendo que solo el jugador de referencia pueda acceder al juego.. Además, se utiliza la biblioteca `speech_recognition` para reconocer la palabra de inicio del juego. Esto brinda una forma adicional de interactuar con el juego, ya que el jugador puede simplemente decir la palabra *Start* en lugar de presionar un botón o tocar la pantalla.

3. Requisitos

El proyecto cumple con los requisitos de HCI (Interacción Humano-Computadora) al permitir al jugador interactuar con el juego utilizando marcadores ArUco y reconocimiento facial y de voz. También utiliza la consciencia de contexto al ajustar la escala y rotación del cubo y al dibujar los checkpoints en los frames detectados.

El juego resuelve un problema no trivial al combinar la realidad aumentada, el reconocimiento facial y la detección de marcadores ArUco para crear una experiencia interactiva y divertida.

La aplicación tiene utilidad real fuera del laboratorio, ya que puede ser utilizada como un juego de entretenimiento y también como una demostración de las capacidades de la realidad aumentada y el reconocimiento facial.

4. Mejoras para la colaboración entre jugadores

Si un jugador es el conductor y otro jugador es responsable de posicionar los marcadores, aquí hay algunas mejoras que podrían abordar problemas triviales adicionales:

4.1. Colaboración y comunicación

Los jugadores deberán comunicarse y coordinarse entre sí para posicionar los marcadores de manera estratégica. Esto promoverá la colaboración y la capacidad de trabajar en equipo, ya que tendrán que discutir y planificar la ubicación óptima de los checkpoints para dificultar la carrera del otro jugador.

4.2. Competencia estratégica

Ambos jugadores estarán involucrados en una competencia estratégica en tiempo real. El conductor deberá anticipar los movimientos del jugador que coloca los marcadores y adaptar su estrategia en consecuencia, mientras que el jugador de los marcadores intentará engañar y dificultar al conductor mediante la ubicación y manipulación de los checkpoints.

Al involucrar a dos jugadores en roles diferentes y permitir la interacción competitiva entre ellos, se pueden abordar problemas triviales adicionales relacionados con la colaboración, la toma de decisiones tácticas, la competencia estratégica y las habilidades de negociación. Esto añadirá un nivel adicional de complejidad y diversión al juego de realidad aumentada.

5. Puntos de la implantación

- **Detección de marcadores ArUco:** Utilizar OpenCV para la detección de marcadores ArUco en tiempo real. Esto implica la captura de video desde una cámara y el procesamiento de los frames para identificar los marcadores ArUco presentes en la escena.
- **Reconocimiento facial:** Utilizar la biblioteca face_recognition para el reconocimiento facial. Esto permitirá identificar al jugador de referencia y otorgarle acceso al juego. El reconocimiento facial se realizará en tiempo real y se comparará con una base de datos de rostros autorizados.
- **Generación aleatoria de checkpoints:** Generar un orden aleatorio preestablecido en el que los checkpoints deben ser recorridos. Este orden se mostrará al inicio del juego y no se volverá a mostrar, lo que requerirá que el jugador memorice el orden de los checkpoints.
- **Interacción con marcadores ArUco:** Permitir al jugador interactuar con el juego utilizando los marcadores ArUco. Esto implica detectar cuando el jugador pasa por un checkpoint al colocar un marcador en la posición correcta en el mundo real. El tiempo se registrará desde el inicio hasta que se hayan recorrido todos los checkpoints en el orden correcto.
- **Interfaz de usuario y pantallas:** Crear una interfaz de usuario que muestre la información necesaria, como el tiempo transcurrido, el orden de los checkpoints que ha visitado y la detección de la palabra *Start* para iniciar el juego. También se deberán diseñar pantallas adicionales, como una pantalla de inicio, una pantalla de victoria y una pantalla de derrota.
- **Implementación de lógica del juego:** Implementar la lógica del juego para determinar si el jugador ha pasado por los checkpoints en el orden correcto y en el menor tiempo

posible. Esto implica comparar el orden de los marcadores ArUco detectados con el orden aleatorio preestablecido y verificar si coinciden.

- **Colaboración en tiempo real:** Si se implementa la colaboración entre dos jugadores, se deberá establecer un sistema de comunicación en tiempo real para permitir que el jugador responsable de posicionar los marcadores interactúe con el conductor. Esto podría involucrar la transmisión de información sobre la ubicación y manipulación de los checkpoints.

Estos puntos clave representan las principales funcionalidades y elementos que se deben implementar en el juego de realidad aumentada utilizando marcadores ArUco y reconocimiento facial. Cada uno de ellos requiere una implementación específica para lograr el funcionamiento completo y la experiencia interactiva deseada.

6. Implementación

Para la implementación del juego seguimos los siguientes pasos:

- Reconocimiento facial: Primero que nada solo podrá acceder al juego el jugador a reconocer
- Inicializamos orden de los marcadores y demás parámetros del juego e iniciamos la cámara
- Como previamente hemos realizado la calibración de la cámara para evitar deformaciones y demás, aplicamos esto conseguimos la nueva cámara con los parámetros de escalado apropiados.
- Una vez que esta todo listo, mediante el reconocimiento de voz esperaremos a que se pronuncie la palabra *START* para comenzar el juego, en este momento comenzara el tiempo de juego.
- Encendemos la cámara y procesamos el frame para corregir la cámara y recortarla, una vez que tenemos esto tenemos la base sobre la que trabajar y procesar los frames recortados.
- El juego ya a comenzado, ahora tendremos 10 segundos para memorizar el orden en el que tendremos que recorrer los marcadores, también podremos pulsar la tecla *s* si ya hemos memorizado el orden y queremos empezar, esto nos ahorra tiempo pero tendrás que decidir si has sido capaz de memorizar todos los puntos.
- Cuando ya has pasado de esa fase de muestra de información ahora llega el procesado de imágenes en si.
 - Detectamos los marcadores ArUco de los frames
 - Calculamos sus posiciones, es decir sus traslaciones y rotaciones tal y como se detectan en la cámara.
 - Una vez que tenemos eso podemos empezar a añadir funcionalidades:

- Dibujamos al coche de Mario en la posición correcta

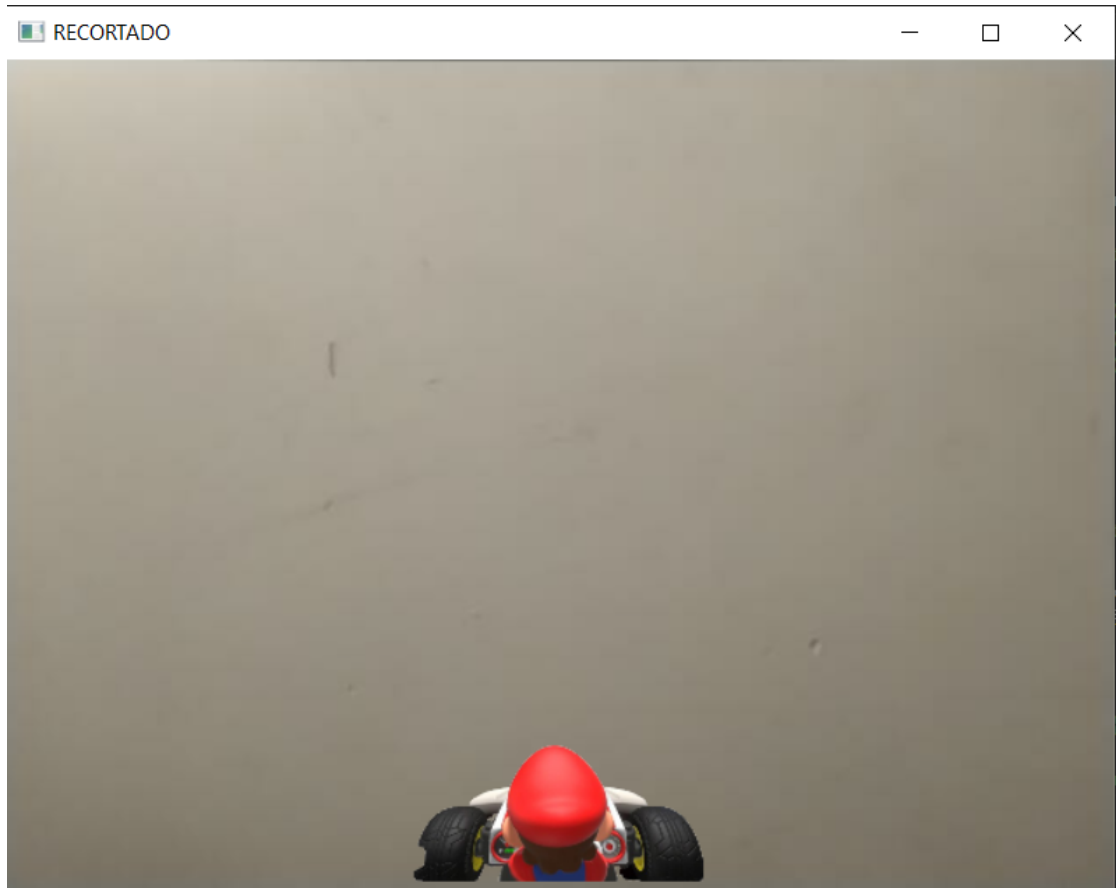


Figura 6.1

- Calculamos la distancia del coche hasta el marcador y la mostramos sobre una línea que conecte estos dos puntos.

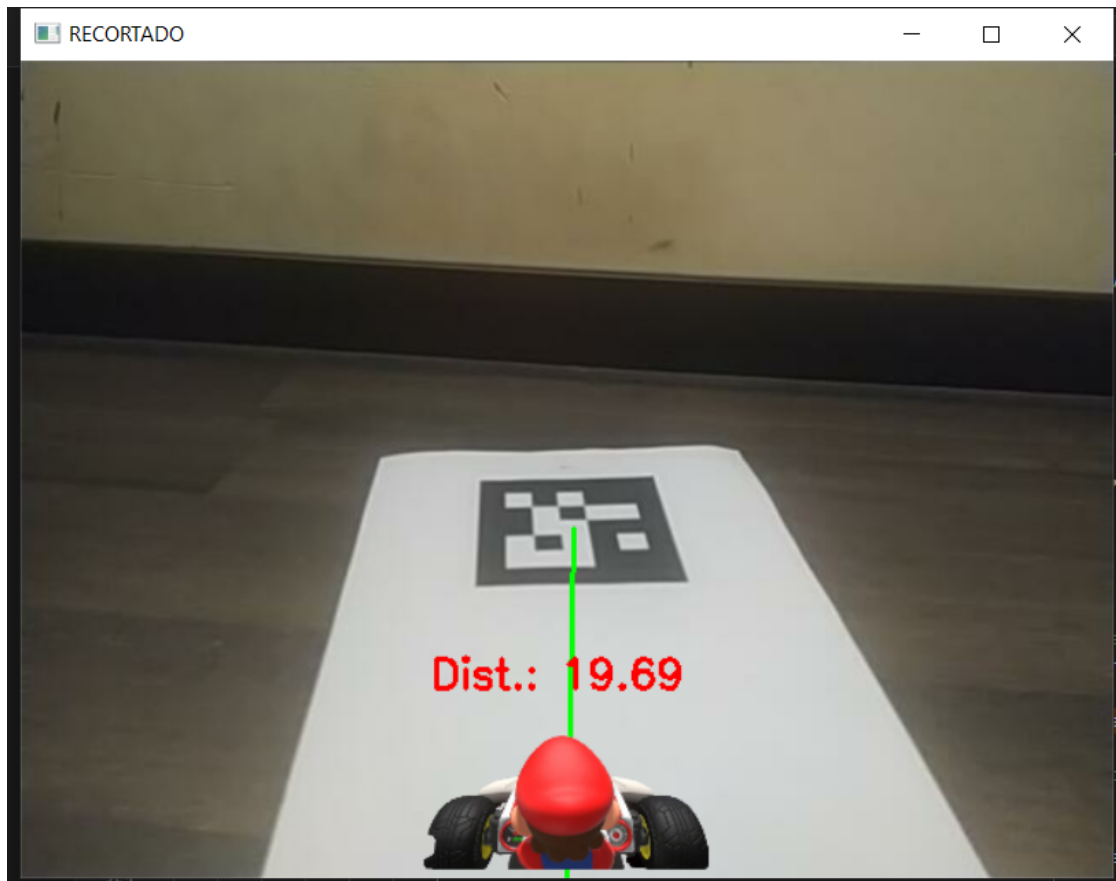


Figura 6.2

- Dibujamos los checkpoints

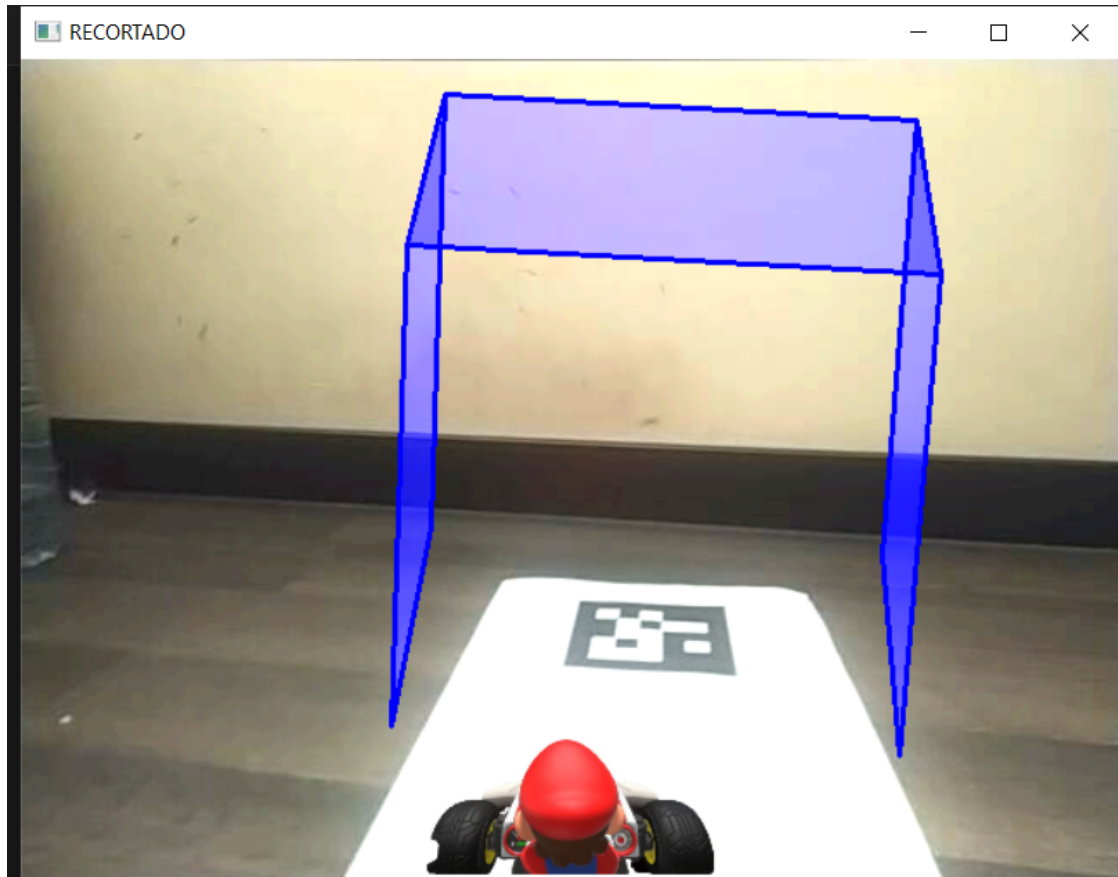


Figura 6.3

- Dibujamos los cubos rotándose

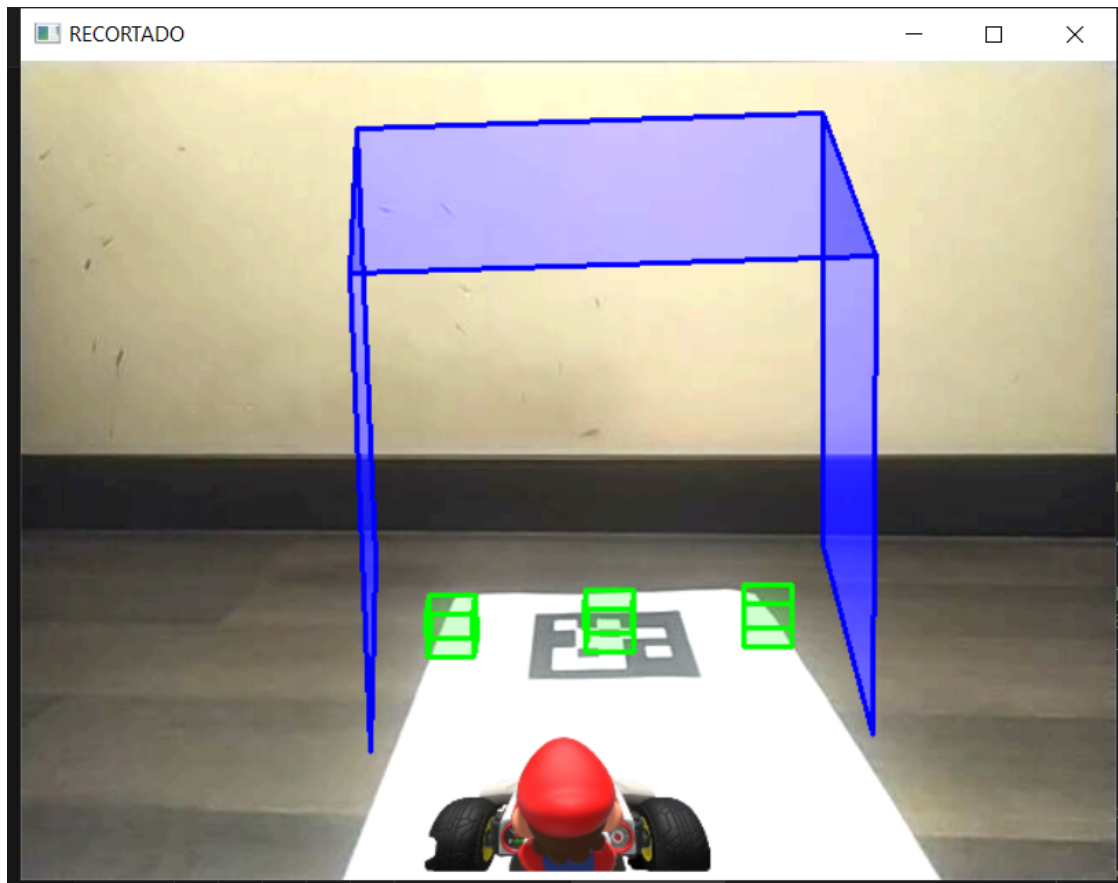


Figura 6.4

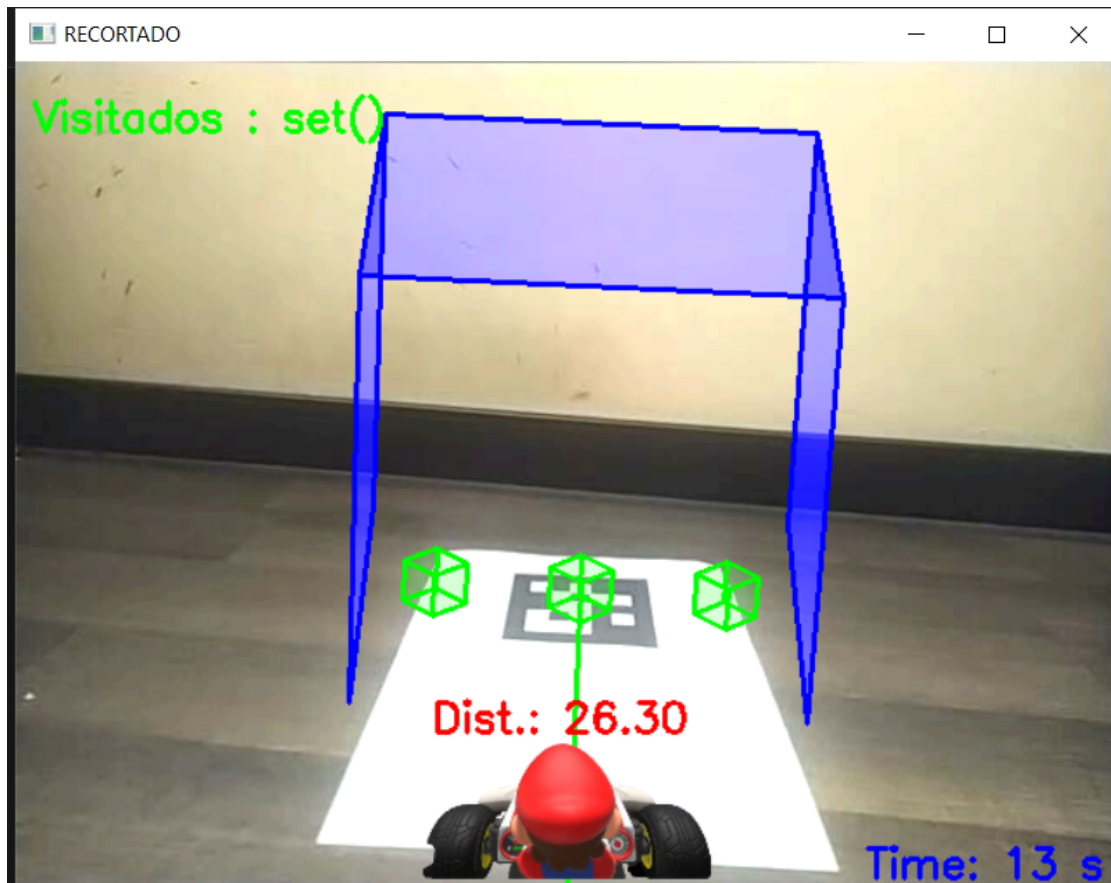


Figura 6.5

- Durante el juego, en la interfaz se mostrara toda esta información ademas de el tiempo actual de juego, junto al numero y orden de checkpoints visitados
- Una vez que se complete el juego se mostrara una pantalla de victoria con el tiempo que nos ha tomado resolver el juego.

7. TODO

- Agregar efectos visuales y de sonido para hacer el juego más inmersivo y emocionante. Por ejemplo, se pueden añadir explosiones o efectos de velocidad cuando se pasa por un checkpoint.
- Implementar niveles de dificultad que ajusten la velocidad del juego o la cantidad de checkpoints a pasar en un tiempo determinado.
- Permitir la opción de jugar en modo multijugador, donde varios jugadores compitan en tiempo real por pasar los checkpoints en el orden correcto.

- Agregar power-ups o elementos interactivos en el entorno de juego que afecten la velocidad o las habilidades del jugador.
- Incluir un sistema de puntuación y clasificación global para motivar a los jugadores a mejorar sus tiempos y competir con otros usuarios.

8. Consciencia de Contexto

Para aplicar eso vamos a necesitar almacenar cierta información de los jugadores. En mi caso se ha desarrollado un archivo *jugadores.json* el cual almacenará dichos datos.

La estructura de los jugadores sera la siguiente:

```
1  [  
2    {  
3      "nombre": "",  
4      "preferencias": {  
5        "idioma": "",  
6        "personaje": ""  
7      },  
8      "historial": [],  
9      "encoding_cara": [],  
10   },  
11 ]
```

De esta forma guardaremos los jugadores por nombre, tendremos también información preferente de dicho jugador como su idioma, personaje preferido para jugar y demás cosas que se podrán añadir. Ademas tendremos un historial del tiempo que ha tardado el jugador en resolver los anteriores juegos asi de estas forma poder estimar el nivel de dicho jugador.

Por ultimo tenemos un campo que es el encoding de la cara el cual es imprescindible y muy util para identificar al jugador.

Gracias a esta estructura ahora cada vez que queramos jugar simplemente nos sentaremos delante del juego, este intentará reconocer la cara y compararla con las que tiene almacenadas en la base de datos, si detecta una coincidencia el sistema sera capaz de identificar al jugador con sus características lo cual hace que el usuario no tenga que hacer nada que le distraiga del juego. Por otro lado si el sistema no es capaz de encontrar un face_encoding para ese jugador determinará que se trata de un jugador nuevo por lo que tendrá que realizar registro del jugador.

Este registro procede de la siguiente forma.

- Obtenemos su face_encoding
- Mediante reconocimiento de voz
 - Preguntaremos su nombre y el jugador nos responderá diciendo su nombre.

- Preguntaremos el personaje con el que quiere jugar y nuevamente responderá con uno dentro de los disponibles

Con estos datos ya somos capaces de estructurar a un jugador y simplemente lo añadiremos a la base de datos.

9. Ampliación Reconocimiento de Voz

Como ya he mencionado antes se ha utilizado reconocimiento de voz para

- Registrar datos de usuario
- Dar el inicio del juego.

Ahora he querido añadir una funcionalidad mas, esta se trata de reconocer el habla mientras se esta jugando, esto puede tener varias funcionalidades y ventajas para el juego. En mi caso yo la he utilizado para poder cambiar el personaje con el cual se esta corriendo.

Para ello simplemente pronunciar el personaje y automáticamente cambiará el diseño. También se podrían escribir anotaciones para ayudar a resolver el juego y muchas mas aplicaciones.

También es util ya que si implementamos una serie de habilidades recolectables en los cubos de los checkpoint podemos usarlas mediante voz de la misma manera, pronunciando "Actíivate" el nombre de la habilidad, para la demo solo se dispondrá de un turbo.