



UNIVERSIDAD DE GUANAJUATO
CAMPUS IRAPUATO – SALAMANCA
DIVISIÓN DE INGENIERÍAS

VISIÓN POR COMPUTADORA
DOCTORA MARÍA SUSANA ÁVILA GARCÍA

ALUMNA DOLLY GWENDOLINE LÓPEZ ROSAS

CORREO dg.lopezrosas@ugto.mx

NUA 379969

PROYECTO FINAL DOCUMENTACIÓN
“CAZADOR DE MONSTRUOS DE COLORES”

Introducción

Cazador de Monstruos de Colores, es un juego interactivo pensado para niños, donde combinamos detección de colores en tiempo real con un pequeño reto visual. La idea era crear algo divertido y educativo al mismo tiempo, utilizando herramientas de visión por computadora y programación.

El juego consiste en mostrar tarjetas de colores frente a una cámara para cazar monstruos que coincidan con el color que el sistema te pide.

Desarrollo

El desarrollar este proyecto se hicieron varios pasos:

1. Planeación

Pensé en un juego que fuera interactivo y educativo para niños, pero que también incluyera detección de colores como parte del reto. La idea principal era cazar monstruos de colores usando tarjetas físicas que los jugadores mostrarían frente a una cámara. Esto hizo que el proyecto fuera una mezcla de visión por computadora, diseño de interfaces y lógicas de juego.

2. Preparación del entorno

Configuramos un entorno de desarrollo usando Python y Flask para el backend, y HTML, CSS y JavaScript para el frontend. Usamos OpenCV para procesar imágenes en tiempo real y detectar colores, ya que esta librería tiene herramientas muy buenas para trabajar con videos y cámaras. Organizamos el proyecto en varias carpetas:

- **static:** Aquí colocamos los sonidos (correct.mp3 y wrong.mp3) y las imágenes de los monstruos.
- **templates:** Aquí se almacenó el archivo index.html para la interfaz del juego.
- **Server.py** Este archivo controla toda la lógica del backend, incluyendo la cámara y el envío de datos al navegador.

3. Lógica del juego

Generar monstruos: Creamos una función que coloca de forma aleatoria entre 3 y 5 monstruos en la pantalla. Uno de estos monstruos siempre coincide con el color objetivo, mientras que los otros tienen colores diferentes. Esto lo logramos usando un arreglo de colores (["ROJO", "AZUL", "AMARILLO", "VERDE", "MORADO"]) y seleccionando aleatoriamente uno como objetivo.

Cambio dinámico de monstruos: Cada vez que el jugador acierta mostrando el color correcto, los monstruos se regeneran con nuevos colores, y el color objetivo también cambia.

Sistema de puntuación: Por cada color correcto, el jugador gana 10 puntos. Los puntos se actualizan en tiempo real en la pantalla.

4. Implementación de detección de colores

Este fue uno de los retos más importantes del proyecto. Usamos OpenCV para capturar imágenes en tiempo real desde la cámara y calcular el promedio de colores en una región central. Aquí los pasos principales fueron:

- Capturar un cuadro de video en cada iteración.
- Convertir la imagen a un espacio de color RGB y tomar un área específica en el centro de la imagen para calcular el promedio de los colores.
- Comparar los valores promedio con rangos predefinidos para clasificar el color detectado como rojo, azul, amarillo, verde o morado.

Ajustamos los rangos de colores varias veces, ya que, en algunos casos, como con el morado y el verde, no se detectaban correctamente debido a las condiciones de luz.

5. Diseño de interfaz

La pantalla del juego se dividió en dos partes:

Lado izquierdo: Aquí aparecen el título del juego, el temporizador, la puntuación y el color objetivo que el jugador debe buscar.

Lado derecho: Muestra la cámara en tiempo real para que los jugadores puedan ver cómo se detectan los colores. También tiene un mensaje animado que dice "¡Muestra el color aquí!" para guiar al jugador.

Usamos colores pastel y una fuente amigable para que el diseño fuera atractivo para niños. También añadimos un botón grande y verde para iniciar el juego.

6. Sonidos y efectos:

Para hacer el juego más interactivo, agregamos dos sonidos:

Sonido correcto: Se reproduce cuando el jugador acierta el color. Esto refuerza la acción positiva.

Sonido de error: Se reproduce cuando el color no coincide con el objetivo.

Esto lo implementamos con objetos Audio en JavaScript, que se activan dependiendo de si el color detectado coincide con el objetivo o no.

7. Pruebas y ajustes

Durante las pruebas, enfrentamos varios retos:

Sensibilidad a la luz: Los colores como el morado y el verde no se detectaban bien en condiciones de poca luz, por lo que ajustamos los rangos de RGB para hacerlos más tolerantes. También ampliamos el área de detección de colores para mejorar la precisión.

Retroalimentación: Agregamos mensajes y sonidos para que los jugadores supieran si estaban acertando o fallando.

Velocidad: Ajustamos el intervalo de detección para que el juego fuera lo suficientemente rápido como para sentirlo dinámico, pero no tan rápido como para frustrar a los jugadores.

Resultados

El juego quedó funcional. Lo que más me gustó fue que logramos que detectara colores en tiempo real y que el juego se viera sencillo pero entretenido. Estos son los puntos que considero más importantes del resultado:

Lo que salió bien:

- Detecta los colores básicos (rojo y amarillo) muy bien.
- Es fácil de usar y entender, incluso para niños pequeños.
- Los efectos de sonido hacen que se sienta más dinámico.

Conclusión

Este proyecto me ayudó a poner en práctica lo que aprendimos sobre visión por computadora. Aunque al principio tuve algunos problemas con la detección de colores, fue interesante ir ajustando y probando hasta que quedó bien. También aprendí a combinar varias tecnologías (Python, Flask, JavaScript) para hacer algo funcional.