

Game Detection 2048

FACULTAD DE INGENIERÍA
UNIVERSIDAD DE ANTIOQUIA
Departamento de Electrónica y Telecomunicaciones
Semestre 2021-2

Yeison Monsalve Sanchez

yeison.monsalves@udea.edu.co

Camilo Hernandez Ruiz

camilo.hernandez1@udea.edu.co

Descripción del Problema

- **Problema humano:** Desarrollar la detección en tiempo real de un objeto físico con un entorno computacional por medio de un juego matemático.
- **Problema Técnico:** Identificar por medio de una webcam la posición de un objeto para la interacción con el juego y su funcionamiento.
- **Estado del arte:**
 - Marcador o Lápiz virtual | OpenCV - Python
 - OpenCV Track Object Movement - PyImageSearch
 - (PDF) Object Detection and Tracking Using OpenCV in Python (researchgate.net)

Marco teórico de las técnicas a emplear

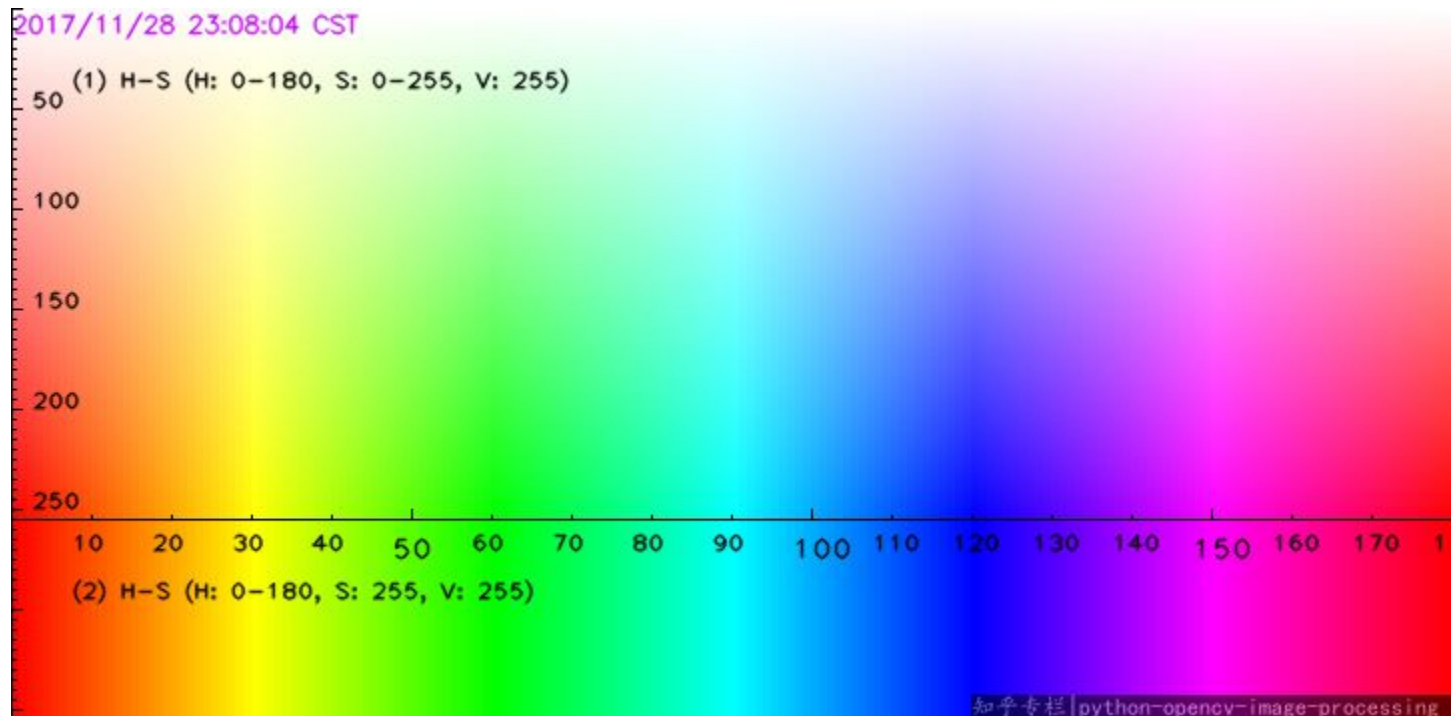
- Técnica 1: Umbralización y binarización para la creación del kernel



- Espacios de colores
- Valores de Umbral
- Binarización según el valor (máscara)

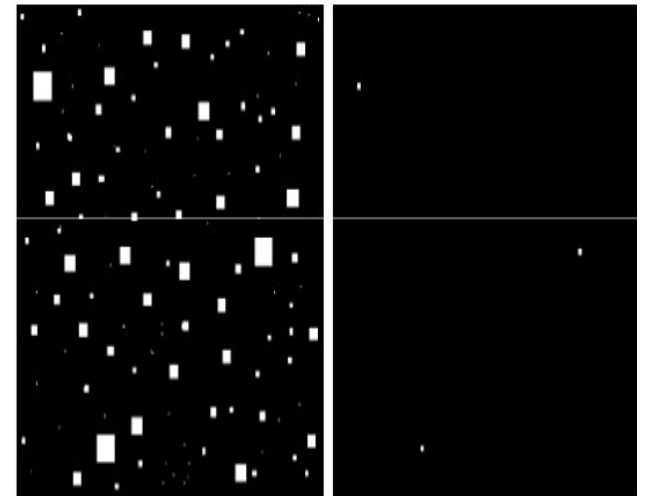
Marco teórico de las técnicas a emplear

- Técnica 1: Umbralización y binarización para la creación del kernel



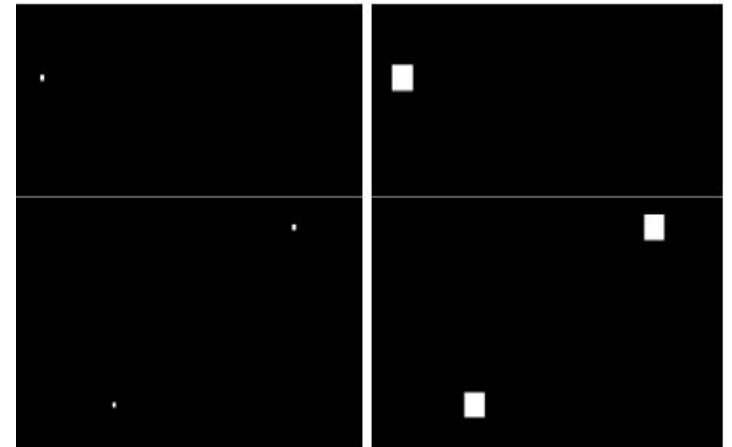
Marco teórico de las técnicas a emplear

- Técnica 2: Transformaciones morfológicas: **erosión**



Marco teórico de las técnicas a emplear

- Técnica 2: Transformaciones morfológicas: **dilatación**



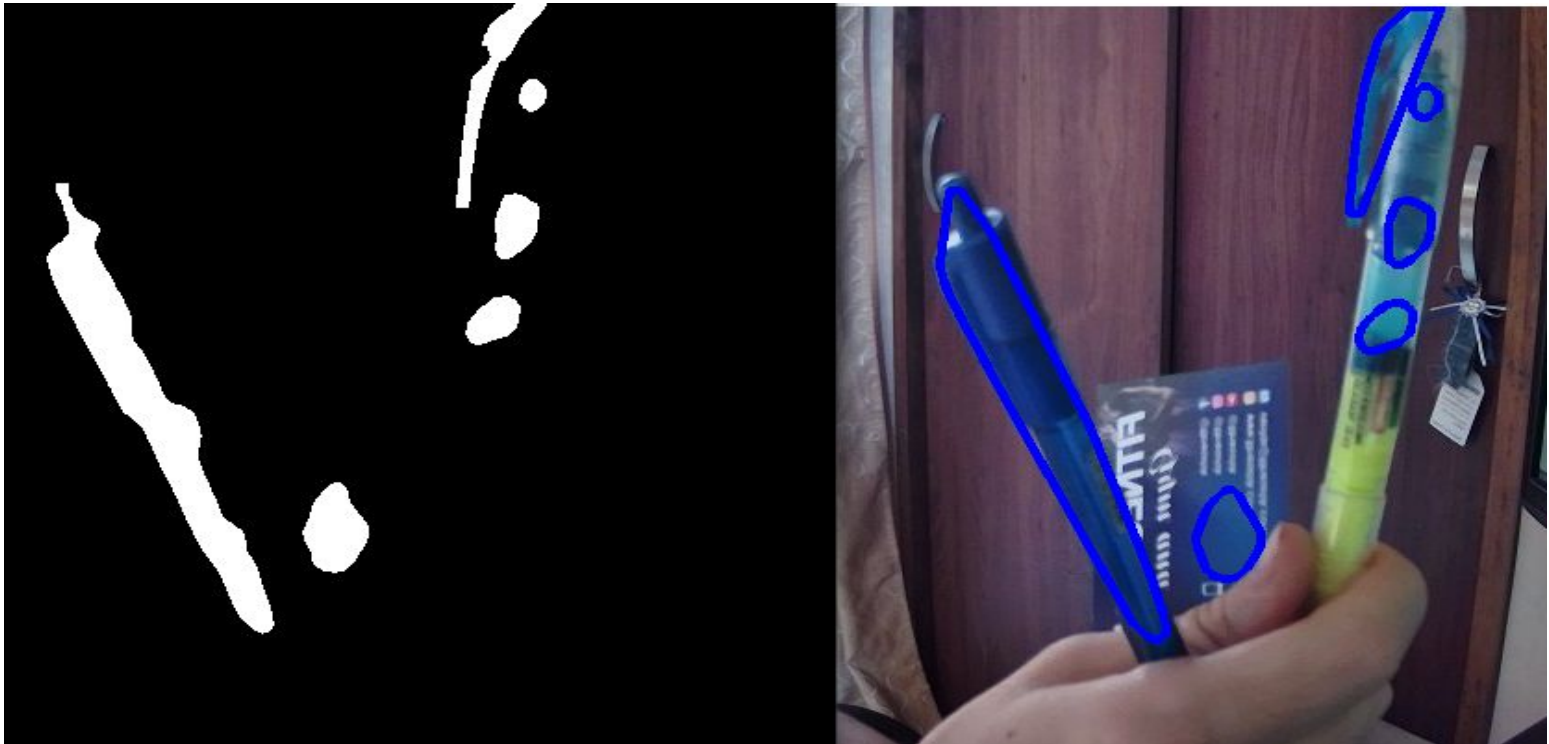
Marco teórico de las técnicas a emplear

- Técnica 2: Transformaciones morfológicas:
difuminación media (*Median Blur*)



Marco teórico de las técnicas a emplear

- Técnica 3: Detección de contornos.



Marco teórico de las técnicas a emplear

- Técnica 3: **Detección de contornos.**

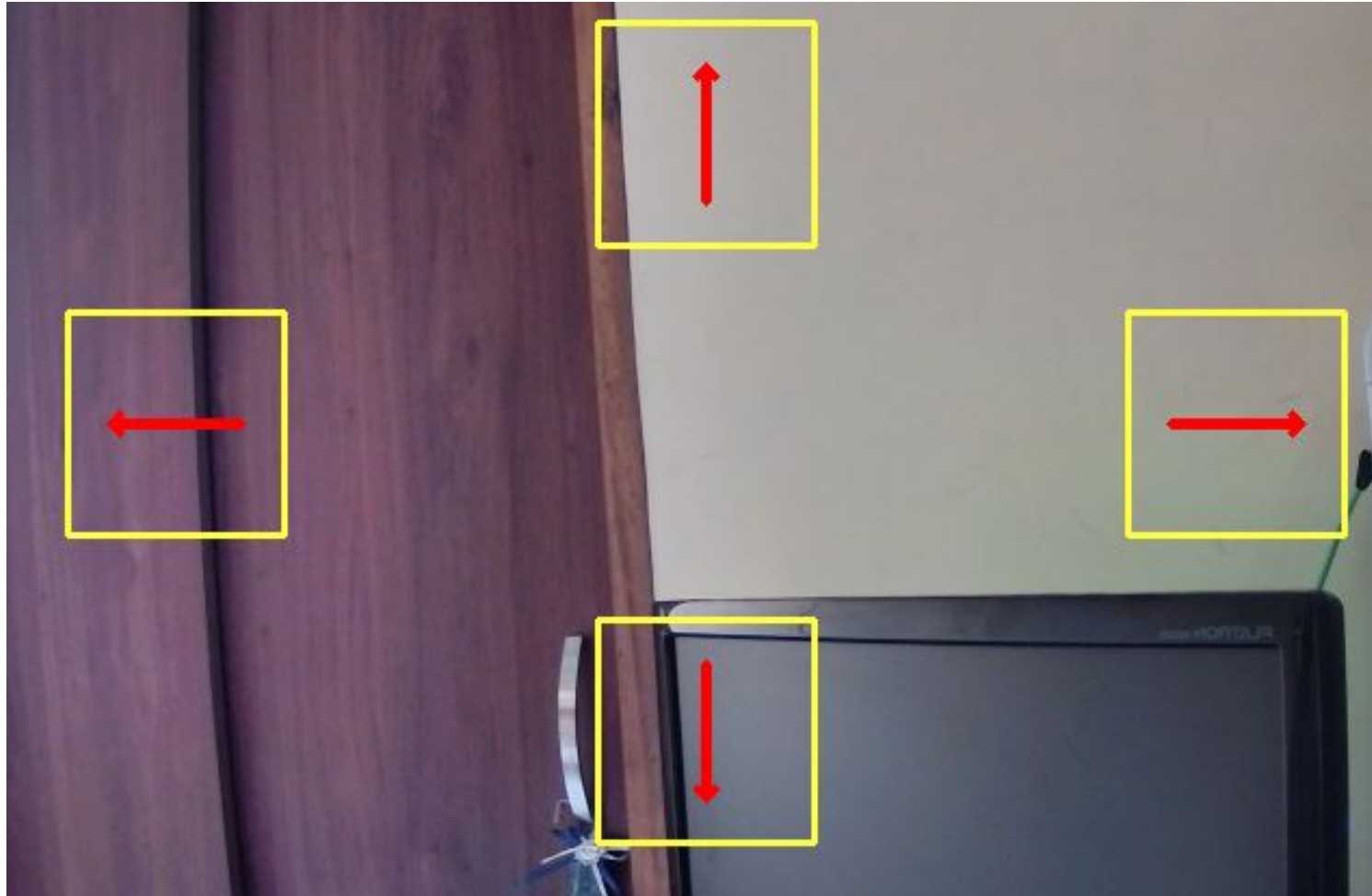


Marco teórico de las técnicas a emplear

- Técnica 4: rectángulo de contorno.

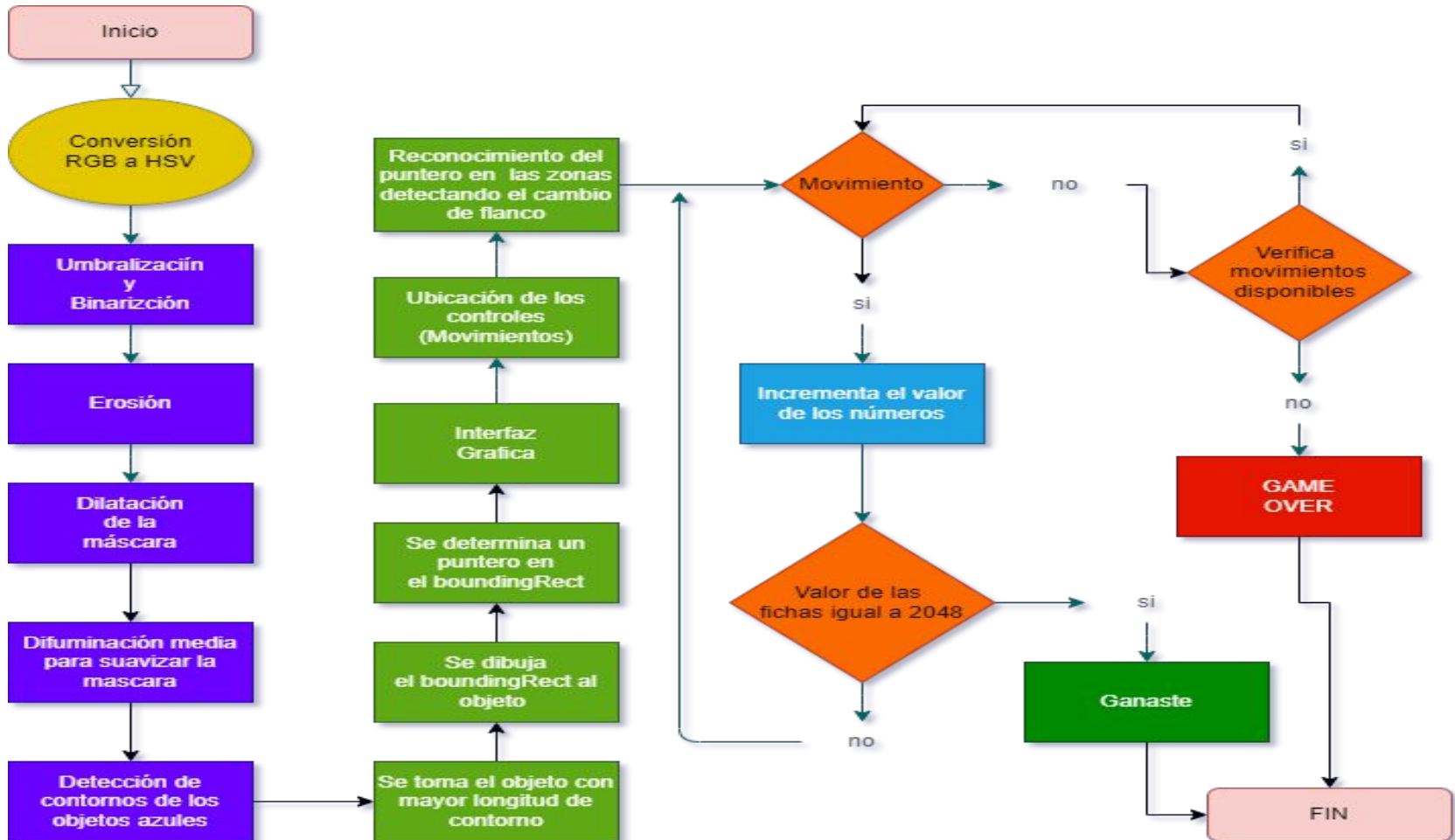


Interfaz para el control del Juego

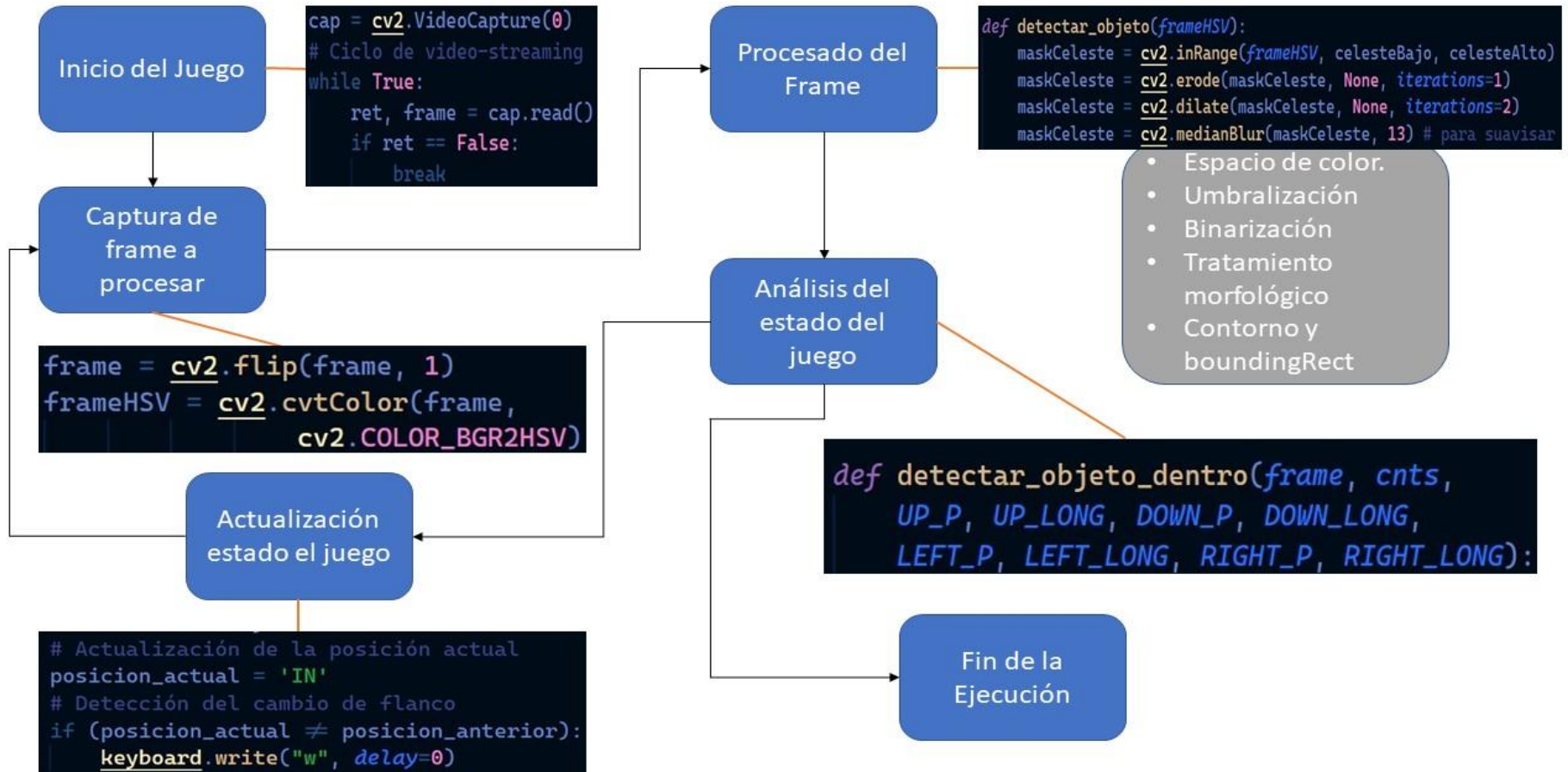


Propuesta de Solución

Diagrama de Flujo



Estructura del Código



Resultados y líneas futuras

- Se hizo más fácil el reconocimiento del objeto cuando se hace una representación del frame en el espacio de color HSV; esto hizo posible una mejor detección del color que permitió la detección del contorno del objeto y por ende, una mejor aproximación de la posición de éste dentro de las dimensiones del frame.
- El porcentaje de error al momento de detectar el lapicero azul no es 100% preciso. Esto se nota al ver que el puntero sobre el lado superior del rectángulo delimitador es parpadeante o salta mucho entre píxeles, lo que implica, en términos del detector, que el usuario está moviendo los números del juego (presionando los botones) aún sin mover el lapicero. En pocas palabras, es muy sensible a los pequeños cambios de luz a baja intensidad.
- Líneas futuras:
 - Detección del objeto aún cuando este se desplace rápidamente.
 - Incrementar los niveles del juego, para cuando complete los 2048 puntos, pregunte al usuario si quiere ir a por los 4096.

Bibliografía y webgrafía

- **[1]** Procesamiento Digital de Imágenes y Reconocimiento de Patrones, 05_Umbral_Histograma.pdf, David Fernández Mc Cann, <https://drive.google.com/file/d/0B7iq2X4FMSZIM2xSY0l5aHN3c1k/view>
- **[2]** Tutorial de procesar imágenes en Python (usando OpenCV), likegeeks.com, recuperado de: <https://likegeeks.com/es/procesar-de-imagenes-en-python/>
- **[3]** Object detection and tracking using OpenCV and Python, programmer Sought, recuperado de: <https://www.programmersought.com/article/9739995566/>

Preguntas del público?

Aplausos?...