

HandCursor

Jason Steven Estrada Russill

Laura Sofia Gomez Rosales

Universidad del Norte

Programación Orientada a Objetos NRC 3492

Ing. Jose Posada Aguilar

2022

Introducción

La tecnología avanza e innova con la intención de facilitar nuestra vida diaria. Por ejemplo, cierto día tienes una presentación importante y ya tienes la presentación proyectada al tablero, pero olvidaste el control remoto que te permite controlar la presentación a distancia. En vez de acercarte al PC para cambiar de diapositiva, puedes abrir el software Hand Cursor, que tiene como propósito agilizar ciertos comportamientos al momento de usar la computadora, reemplazando al mouse físico e integrando algunas funcionalidades propias del teclado para: adelantar o retroceder diapositiva, mover el cursor por la pantalla, hacer clic sobre los botones, etc. Todo esto desde la distancia, ya que su utilidad consiste en la detección de los gestos de la mano humana a través de la cámara en tiempo real, así poder efectuar ciertas tareas gracias a librerías y módulos entrenados como OpenCV, CVzone, MediaPipe y PyAutoGUI.

Referentes de soluciones existentes

Al realizar la investigación sobre soluciones ya existentes al problema del control de diapositivas a distancia, se encontró que la mayoría de resultados arrojados son de proyectos independientes y no patentados desarrollados por programadores. A continuación se presenta una lista con las alternativas para el usuario:

1. Microsoft(s. f.) permite utilizar un smartphone como puntero láser y pasar las diapositivas al efectuar una presentación de PowerPoint.

Limitaciones: Funciona exclusivamente con Microsoft office, se debe establecer una conexión del celular al proyector, si la conexión es inalámbrica, es necesario tener conexión a internet.

2. Desde 2017, la empresa DinoFire ha lanzado al mercado de Amazon distintos diseños de controles para diapositivas bajo la marca HEYLACOLX. Al ser un dispositivo enfocado principalmente a la solución de este problema, su eficiencia es alta.

Limitaciones: Requiere mantener el dispositivo con batería, se puede perder el control, el láser puede no ser notorio a distancia.

Soluciones no tan a fines con la situación problema:

3. Mixage Software desarrolló en 2003 Mouse Emulator, a pesar de su fecha de lanzamiento, aún continúa siendo de utilidad porque permite controlar el mouse a través del teclado.

Limitaciones: Sólo funciona para Windows, según la página de descarga de Softonic tiene 3.1 /5 de calificación

4. Oculus Quest es un dispositivo de realidad virtual desarrollado por Meta empleando el Hand Tracking para hacer la experiencia VR más real

Limitaciones: Está relacionado con el proyecto como referentes de implementación del Hand Tracking, pero no es a fin con la problemática.

Proyectos independientes por programadores:

5. MediaPipe Hands: On-device Real-time Hand Tracking (Zhang et al., 2020)

Limitaciones: Proyecto enfocado a la construcción de un módulo funcional para demostrar con la librería MediaPipe el alcance del Hand Tracking

6. Real-time computer vision with OpenCV. (Pulli et al., 2012)

Limitaciones: Proyecto enfocado en demostrar con la librería OpenCV el alcance de la visión por computador

7. Repositorios de Github relacionados con la situación problema desarrollado por programadores independientes: [Virtual Mouse](#)

Alcance

Hand Cursor, hasta la fecha del 29 de mayo del 2022, está diseñado para ejecutar con éxito las siguientes funciones desde la distancia:

- Mover el cursor sobre la pantalla
- Hacer clic derecho e izquierdo
- Efectuar las funciones de las teclas: arriba, abajo, derecha e izquierda del teclado

Permitiendo el control sobre las diapositivas, de esta forma los expositores no tendrán que acercarse hasta el computador para poder cambiar de diapositiva.

Respecto a la planeación inicial del proyecto con la presentada, estas son las consideraciones que no se lograron implementar a tiempo:

- Ambiente controlado para correcto funcionamiento, ya que a mucha o poca luz, hay mayor margen de error del programa para la identificación de las manos
- Implementar una manilla de color fluorescente para que el programa solo identifique aquella mano que la porte.
- Agregación de más funciones relacionadas con el mouse como: clic sostenido, sostener y soltar, tercer botón, configuración de sensibilidad, etc.
- Adaptación a sistemas operativos como Mac y Linux

Alternativas y escogencia de alternativas

Teniendo en cuenta los referentes existentes como alternativas a la solución de poder controlar las diapositivas de una presentación a distancia, se decidió combinar tecnologías como:

- Lenguajes de programación Python
- El IDE Pycharm Community
- Librería OpenCV para la visión por computadora. Ejemplo: activar la cámara.
- Librería MediaPipe para el seguimiento de la mano
- PyAutoGUI para ejecutar los comandos seleccionados del mouse y teclado

Hand Cursor es diferente a los mouse virtual presentado por desarrolladores en GitHub porque permite realizar gestos con las manos equivalentes a las flechas del teclado, es así como las diapositivas se pueden cambiar. Como extra, por ejemplo, se puede jugar al dinosaurio de Google, ya que también funciona por las flechas del teclado.

Diseño de solución

Requerimientos funcionales

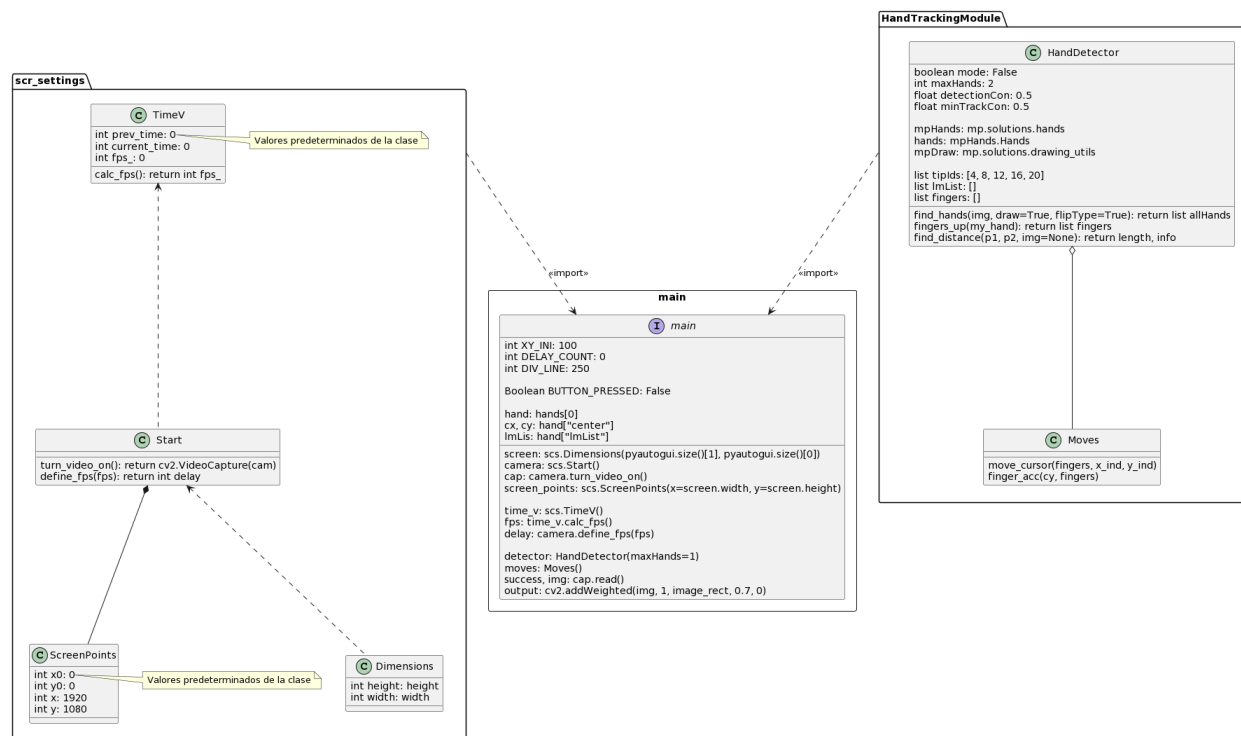
- Identificación de la mano humana a través del visor de la cámara en tiempo real
- Acceso y activación de la cámara conectada al ordenador. Si no tiene cámara interna el PC, ni una cámara externa conectada, no se podrá emplear el programa.
- Interfaz de selección de cámara
- En caso de que la mano con que se haga el gesto le falte uno o más dedos, se limitarán las funciones a las que el usuario puede acceder
- Calcular la distancia entre la mano y la cámara

- Calcular la relación de aspecto entre la pantalla del ordenador y la zona de desplazamiento del cursor sobre el espacio
- Identificar los distintos gestos de la mano humana establecidos por el programa

Requerimientos no funcionales

- Lenguaje de programación: Python
- Software ejecutable en PC
- Interfaz amigable para el usuario
- Proyección de la salida de la cámara
- Instrucciones de uso
- Requerimientos legales
- La aplicación de escritorio debe funcionar en Windows

UML de clases



El diagrama de clases presenta cómo está construido HandCursor.

Cuando el usuario ejecuta el programa y corre la interfaz, se empieza a configurar el software a partir de los requerimientos de las siguientes clases:

El módulo principal es **HandTrackingModule**, ya que mediante la clase *HandDetector* es el encargo de interpretar los 21 puntos de una mano y clasificarlos por tipo de dedo, en listas de acuerdo mediante la función *find_hands*. También está la función *fingers_up*, que retorna que dedo está levantado basándose en los puntos visibles para la cámara conectada al computador, definiendo como 0 para los dedos que estén abajo y 1 uno aquellos que estén levantados. A partir de allí, funciona la clase *Moves*, esta ejecuta cierto comando dependiendo de la señal que realice la mano frente a la cámara.

El módulo *src_settings* contiene la clase *Dimensions* que determina las dimensiones de a proyectar de la cámara y relación de aspecto a partir del tamaño de la pantalla del usuario, de esta forma, el programa se adapta a cualquier pantalla.

La principal clase es *start*, encargada de validar que el usuario tenga una cámara conectada su computador y a partir del cálculo de fps de la función *calc_fps* de la clase *TimeV*, es que se puede determinar el delay con que se leerán los datos por el visor; para esto el usuario podrá escoger entre la opción “Juego” que tiene un delay de 0 o “Diapositiva”, con un delay que corre a la mitad de los fps obtenidos.

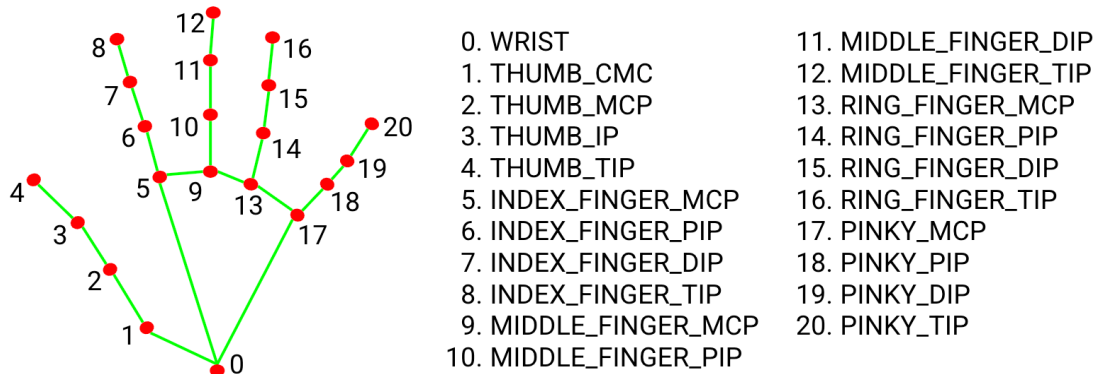
Documentación

Documentación del código:

Hand Cursor está escrito en lenguaje de programación Python, el cual tiene acceso y compatibilidad con las siguientes librerías entrenadas para implementarse de acuerdo a una necesidad. En nuestro caso, ejecutar ciertos comandos a distancia a realizando gestos con la mano hacia el visor de una cámara encendida conectada al computador:

- **OpenCV** es una librería de software de visión por ordenador y aprendizaje automático de código abierto. Este paquete es clave para Hand Cursor, porque es el encargado del procesamiento de imágenes y videos capturados, así mismo permite dibujar líneas, texto y figuras sobre el recuadro que proyecta la imagen resultante de la cámara.
- **MediaPipe** tiene la capacidad de percibir la forma y el movimiento de las manos. El programa emplea el aprendizaje automático para inferir 21 puntos de referencia en 3D de una mano a partir de un solo fotograma. La librería continua en entrenamiento, es por esto que aún su margen de error no es el óptimo.

Teniendo en cuenta los 21 referenciales de la mano establecidos por MediaPipe, se importó y modificó el módulo HandTrackingModulo de CVZone, permitiendo en listar los puntos pertenecientes a cada dedo, tomando como punto inicial el punto 0 y el final el de la punta del dedo. A partir de lo anterior se determina cuando un dedo está arriba o abajo. Ejemplo: el dedo meñique está compuesto por los números [0, 17, 18, 19, 20], si todos estos puntos son visibles para la cámara, se registrará con valor de 1, es decir, el dedo está arriba. Por el contrario, si los puntos intermedios no son visibles y el punto 20 está más cerca del punto 0, se determina que el dedo está abajo, dando como resultado 0.



- PyAutoGUI** permite que los scripts de Python controlen el mouse y el teclado para automatizar las interacciones con otras aplicaciones. Sus métodos y funciones se emplean cada vez que cierta condición se cumpla. Ejemplo: Si el dedo índice y medio están levantados, se calcula las coordenadas X y Y a las cuales el cursor se debe desplazar por la pantalla del PC
- Numpy** es el paquete fundamental para la computación científica en Python, debido a que proporciona un objeto de matriz multidimensional y varios objetos derivados (como matrices y matrices enmascaradas). Gracias a su función de interpolación lineal, `interp()`, permite relacionar dos dimensiones similares por medio de coordenadas. En el proyecto se hace una relación entre las coordenadas de la pantalla del computador y la zona predeterminada en que se desplaza la mano cuando se ejecuta el comando de mover el cursor.

Manual de usuario

- El programa es código abierto, estos son los pasos para clonar el repositorio para usuarios

Windows:

- Dirígete al repositorio de Github: [HandCursor](#)
- Para un mejor desempeño, se recomienda instalar el IDE [Pycharm Community](#)







3. Para clonar el repositorio, se debe dar clic en el botón verde y copiar el link del repositorio
4. De nuevo en Pycharm, da clic en VCS
5. Clic en control de versiones
6. Pega el enlace y elige el directorio
7. Clic en clonar
8. Las librerías a instalar estarán subrayadas, al posarte sobre ellas aparecerá el botón de instalar. Se recomienda primero instalar mediapipe, ya que esta instala automáticamente OpenCV, numpy, math y time. Luego instalar PyAutoGUI
9. Luego de instalar las librerías, ya se puede ejecutar el programa



II. Lista de funcionalidades

En la siguiente tabla se muestran los gestos y la función que realizan únicamente con una mano.

Se debe tener en cuenta que los dedos de una mano se interpretan mediante una lista de 5 espacios, donde:

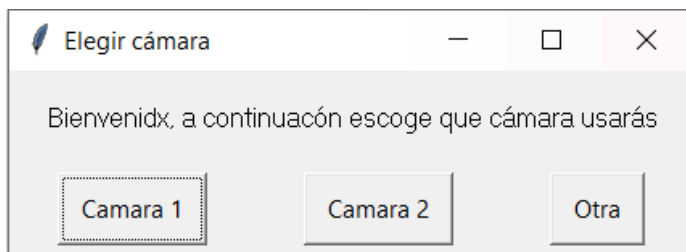
- 1: dedo levantado
- 0: dedo abajo
 - 1ra posición de la lista: pulgar
 - 2da posición de la lista: índice
 - 3ra posición de la lista: corazón/medio
 - 4ta posición de la lista: anular
 - 5ta posición de la lista: meñique

Dedos levantados		Función
[0,1,0,0,0]		Mover el cursor
[0,0,0,0,1]		Clic izquierdo
[0,0,0,0,0]		Clic derecho
[1,1,1,1,1]		Soltar el cursor
[1,0,0,0,0]		Presionar tecla "flecha izquierda"
[0,0,1,1,1]		Presionar tecla "flecha derecha"

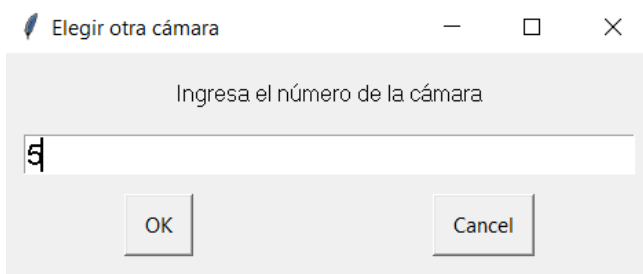
[1,1,1,0,0]		Presionar tecla "flecha arriba"
[0,1,1,0,0]		Presionar tecla "flecha abajo"

III. Elementos de la interfaz

Al ejecutar el programa aparecerán los siguientes cuadros:



Por defecto se muestran 3 botones para escoger la cámara que se usará para procesar las imágenes y ejecutar los comandos. Cámara 1 y 2, son predeterminadas y el botón *Otra* es

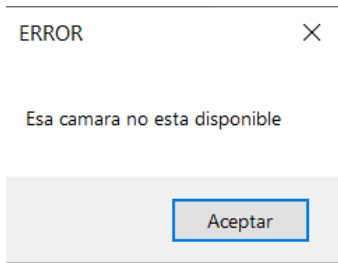


para ingresar una cámara distinta. Si le da clic, se verá de la siguiente manera:

Escribe el dígito de la cámara a utilizar. *Ok* para confirmar y *Cancel* para retroceder.

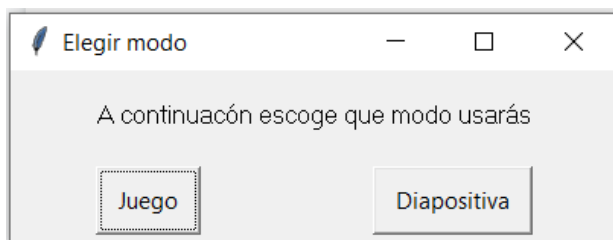
Sí se selecciona una cámara inasequible, se muestra el siguiente mensaje de error:

Para volver a iniciar el programa, se debe cerrar con el botón *Aceptar* o en la X



Si el HandCursor se ejecuta sin problema, se mostrará en pantalla dos modos de uso:

Juego y Diapositiva, la diferencia entre los dos es el delay con que se procesaran las imágenes y, por lo tanto, la velocidad que se realizará un comando.



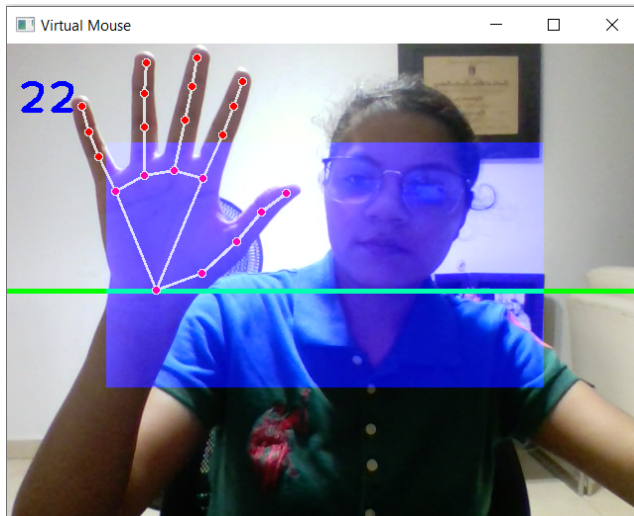
A partir de aquí, se puede disfrutar de la experiencia de HandCursor y sus funcionalidades descritas anteriormente.

La zona azul relaciona las coordenadas pantalla del PC con las indicadas por el índice de la mano al mover el cursor.

La línea verde indica que solo se leerán los comandos por encima de este límite. No aplica para mover el cursor.

En la esquina superior izquierda, se muestra a cuantos fps se está leyendo la imagen.

Para salir del programa, se debe dar clic sobre el cuadro de la app y presionar la tecla *Esc*.



Evidencia de los roles

Fecha	Integrante	Notas
06/04/2022	Estrada y Gómez ▾	Planteamiento inicial del diagrama de clases para el módulo HandTrackingModule y el programa principal VirtualMouse
18/04/2022	Estrada y Gómez ▾	Segundo planteamiento del UML y primer reporte del proyecto con la información de: Planteamiento del problema, requerimientos funcionales y no funcionales, alternativas de solución y tecnologías a implementar
21/04/2022	Laura Gómez ▾	Creación y actualización del Github con la documentación del código
21/04/2022	Jason Estrada ▾	Modificación del módulo HandTrackingModule de CVZone para identificar correctamente las manos del usuario
21/04/2022	Laura Gómez ▾	Primer script del programa adaptado a las medidas de pantalla, resolución de cámara, software y hardware de un computador Asus core i5. Este script se llama Draft.py
27/04/2022	Laura Gómez ▾	Realización de pruebas de campo del archivo Draft.py en el bambú del K con los amigos de Laura Gómez. Se determinó que a mucha y poca luz el margen de error para identificar las manos, aumenta.
28/04/2022	Laura Gómez ▾	Realización de pruebas de campo en el salón 31J del archivo Draft.py con la profesora de inglés Hillary Wade. Se determinó, que el programa cambia de diapositiva muy rápido para usarlo en una clase.
04/05/2022	Laura Gómez ▾	Esqueleto de las clases a implementar en Python. Este módulo se llamó VirtualMouse.py En esta versión el programa enciende la cámara e identifica las manos.
15/05/2022	Jason Estrada ▾	Modificación del Draft.py para definir la altura a la cual se deben hacer los gestos de las manos y realizar los comandos
18/05/2022	Jason Estrada ▾	Adaptación de los frames de relación de aspecto entre

Fecha	Integrante	Notas
		la cámara y la pantalla del computador. Esta modificación se hizo en el script Draft.py
23/05/2022	Jason Estrada ▾	Modificación del esqueleto de clases del módulo VirtualMouse.py utilizando como guía el Draft.py. Esta versión funcional cumple con lo básico que exige la situación problema.
26/05/2022	Laura Gómez ▾	Modificación del módulo VirtualMouse.py Esta versión funcional cumple con lo básico que exige la situación problema
31/05/2022	Laura Gómez ▾	Organización del código en paquetes y módulos.

Conclusiones

Durante el desarrollo de HandCursor hubo diversos problemas, tales como el uso de tecnologías que aún se encuentran en entrenamiento como Mediapipe, ocasionando conflictos a la hora de identificar las manos y que aún no permite hacer archivos ejecutables .exe. Otro reto fue encontrar y modificar un módulo que clasificara los dedos de la mano para que los gestos se identifiquen con precisión, en este caso, empleamos el desarrollado por CVZone. La interpolación lineal llevó tiempo en este proyecto, porque se encarga de establecer la relación de coordenadas entre el área marcada en azul que aparece en la ventana y la pantalla del PC. Dicha interpolación fue encontrada empleando fórmulas matemáticas y el método *interp()* de la librería Numpy. Por último, fue hacer la aplicación adaptable a cualquier equipo con sistema operativo Windows, y esto se consiguió gracias a PyAutoGUI, que obtiene las dimensiones de la pantalla y la cámara seleccionada para ejecutar el programa.

Cabe recalcar que para el óptimo funcionamiento de la aplicación se recomienda estar entornos con buena iluminación, que el usuario no se encuentre a más de 2 metros de la cámara y que tal cámara tenga una resolución que permita que las manos se vean bien enfocadas.

Este proyecto contribuyó al conocimiento de los desarrolladores con el uso de diferentes estructuras pertenecientes al paradigma de la programación orientada a objetos, como clases, objetos, métodos, paquetes, etc. Se aprendió a hacer uso de diferentes librerías enfocadas a la visión por computador y entrenamiento de IA. Finalmente, esta aplicación puede ser mejorada agregando más gestos para ejecutar comandos como click sostenido, hacer scroll o click central.

Referencias

Build software better, together. (s. f.). GitHub. Recuperado 29 de mayo de 2022, de

<https://github.com/search?q=virtual+mouse> (En línea)

CVzone (2022, 8 febrero). *CVzone Github*. CVzone. <https://github.com/cvzone/cvzone> (En línea)

DinoFire. (2017, 15 septiembre). *HEYLACOLX*. Amazon.com. <https://cutt.ly/dJiV2gb> (En línea)

Google (s. f.). *MediaPipe Hands*. MediaPipe. <https://google.github.io/mediapipe/solutions/hands>

Meta. (s. f.). *Meta Quest*. Amazon.es. <https://cutt.ly/eJiV4zV> (En línea)

Microsoft. (s. f.). *Usar un puntero láser en su smartphone durante una presentación de PowerPoint*.

Recuperado 29 de mayo de 2022, de <https://cutt.ly/qJiV5AH> (En línea)

Mixage Software. (2003, 21 febrero). *Mouse Emulator*. Softonic. <https://mouse-emulator.softonic.com/>

NumPy documentation — NumPy v1.22 Manual. (s. f.). Numpy.org. Recuperado 29 de mayo de 2022, de

<https://numpy.org/doc/stable/> (En línea)

OpenCV Team (s. f.). *About*. OpenCV. <https://opencv.org/about/> (En línea)

PyAutoGUI (s. f.-b). *Welcome to PyAutoGUI's documentation!* PyAutoGUI.

<https://pyautogui.readthedocs.io/en/latest/> (En línea)

Pulli, K., Baksheev, A., Korniyakov, K., & Eruhimov, V. (2012, 4 abril). *Real-time computer vision with*

OpenCV. *Communications of the ACM*. ACM. <https://dl.acm.org/doi/10.1145/2181796.2206309>

Zhang, F., Valentin Bazarevsky, V., & Vakunov, A. (2020, 18 junio). *MediaPipe Hands: On-device Real-time*

Hand Tracking. arXiv.Org. <https://arxiv.org/abs/2006.10214> (En línea)