

# Red Glove, herramienta basada en los juegos de encaje hecha en Python con OpenCV2 y PyQt5 para la incorporación de las TIC's en el desarrollo de habilidades motrices

Fuentes Paredes Nelson Alejandro  
Escuela Profesional de Ingeniería de Sistemas  
Universidad Nacional de San Agustín  
Arequipa, Perú  
nfuentesp@unsa.edu.pe

**Abstract - Red Glove es un software que simula los juegos didácticos de encaje de modo gráfico con una interfaz gráfica sencilla, sin embargo, la interacción con el usuario está hecha para que el juego sea controlado con la mano sin necesidad de otros dispositivos de entrada, este tipo de herramientas podría ser ayuda para el desarrollo de las habilidades motrices.**

## I. INTRODUCCIÓN

La llegada de la tecnología ha ayudado a la incorporación de personas con discapacidad en la sociedad, a pesar de eso estas tecnologías, en su mayoría, se concentran meramente en la incorporación, pero no a la prevención.

Red Glove se propone como una herramienta auxiliar en el campo de la estimulación temprana, que ayude a desarrollar habilidades motrices, de un modo novedoso y llamativo para los niños.

Red Glove es un juego basado en los juegos de encaje, este consiste en arrastrar una ficha a una posición que contenga una campo con la misma forma; esto sería un simple juego de arrastre con el mouse, de no ser porque este juego es controlado directamente por la mano, es decir, que la interacción se daría entre el usuario que la juegue y lo que él puede ver en el monitor sin la necesidad de usar el mouse o teclado.

En la sección II se hablara de trabajos similares a este, de la III a la V se mencionara conceptos que pueden ser importantes o relevantes en el tema así como la explicación de métodos y librerías utilizados. La parte VI nos explica de una forma más detallada lo que es Red Glove y la VII parte nos hablará de su implementación pasando brevemente por sus versiones anteriores y explicación de su código fuente

## II. TRABAJOS RELACIONADOS

El desarrollo de técnicas de Accesibilidad, que se han vuelto casi obligatorias en la actualidad en el desarrollo de software, ha llevado a varios estudios de lo que es software orientado a personas con discapacidad cognitiva o la prevención de estas discapacidades a través del desarrollo de las habilidades que merman estas discapacidades durante la infancia o niñez.

En el trabajo de Amores y Jádan [1] encuentran que con el desarrollo de la tecnología en diferentes dispositivos como computadoras o Smartphones cualquier persona tiene fácil acceso y las herramientas que facilitan la rutina diaria, sin embargo, a las personas con alguna discapacidad, como la motriz, no les es tan sencillo el acceso a estas herramientas. Proponen un teclado virtual, que a diferencia de otros, tenga un predictor de texto y sea operado a través de comandos de voz o movimiento de ojos. Obtuvieron como resultados que a diferencia de otros teclados virtuales la predicción de texto ayudó a comunicarse más rápidamente; evitando presionar todas las teclas y el que el control sobre la herramienta vaya creciendo acorde lo permite la herramienta permite que el usuario la manipula de una forma mas rapida y eficaz.

Muro, Santana y García [2] explican que enseñar a leer a niños con síndrome de Down es una tarea educativa difícil que requiere técnicas pedagógicas especiales. Tomando en cuenta todas las características de esta enfermedad concluyen que para estos niños el aprendizaje es lento y es necesario seguir un proceso paso a paso. La optimización, a este problema, que proponen es el uso de pantallas digitales en el método Troncoso y del Cerro, auxiliados por tarjetas de imagen-palabra. Se obtuvo resultados favorables a evaluaciones preliminares; y se comprometen en trabajar en un segundo prototipo con el propósito de apoyar el método Troncoso y del Cerro.

Rivera, Quispe y Montalvo [3] nos informan que en el año que hicieron su investigación la Educación Matemática en los países en vías de desarrollo estaba atravesando por una etapa muy difícil reflejándose en los resultados de las Evaluaciones Internacionales de Calidad Educativa que nos muestran el mal estado en que se encuentran los estudiantes

en el área de matemática. Ellos teorizan que esto se debe a que la enseñanza de esta área se realiza en el Perú de una forma muy clásica, provocando que las habilidades necesarias para ese campo no se desarrollan, por ende, sea más difícil de aprender. En este contexto plantean la implementación de tecnología basada en la realidad aumentada basándose en el desarrollo de las inteligencias múltiples de un modo lúdico. Obtuvo como resultados que la enseñanza con el software educativo mejora el interés en las matemáticas, debido a que los alumnos observan e interactúan con los contenidos, a su vez desarrollan habilidades y destrezas, reforzando la inteligencia múltiple que más desarrollada tiene, así como en una muestra inicial de 5 niños estos, lograron una mejora en el interés de las matemáticas mediante este software por el manejo de objetos gráficos.

Díaz et al [4] nos explican que la proliferación de la tecnología ha ayudado a eliminar las barreras de la comunicación ya sean las clásicas como la geografía o el idioma, también barreras más específicas como lo son las discapacidades motoras, sin embargo, estas tecnologías requieren de algún hardware que en muchas ocasiones es demasiado costoso para un individuo promedio. Frente a esta problemática presentan PyMoHa una herramienta de software libre hecho en Python que no requiere de ningún dispositivo adicional, la función de esta herramienta es la de simular las funciones del mouse a través de las pulsaciones. Este software permite que cualquier persona que pueda realizar al menos un movimiento voluntario, tenga la posibilidad de utilizar diferentes aplicaciones a través de pulsaciones sobre un switch.

Caiza y Campoverde [5] detallan que según los últimos estudios realizados por el Consejo Nacional para la Igualdad de Discapacidades (CONADIS), en el Ecuador existen aproximadamente 401.538 personas con discapacidad. Proponen la aplicación de actividades lúdicas mediante el uso de Interfaces Naturales de Usuario, para el desarrollo de la motricidad gruesa en niños con discapacidades motoras. Se logró brindar a los niños una experiencia nueva y motivadora en donde realizaban terapia física y a la vez de manera inconsciente desarrollan su motricidad gruesa de una manera natural mediante juegos que combinan tecnología con ejercicio físico.

Este trabajo se relaciona con los presentados anteriormente en que se busca dar solución o ayudar de alguna manera a tratar con el desarrollo de habilidades motrices; sin embargo, a diferencia de estos en este trabajo se busca crear o desarrollar una herramienta que pueda servir de auxiliar en el

campo de la estimulación temprana.

### III. JUEGOS DE ENCAJE

#### A. Definición

Según Recursos del Aprendizaje [6] los juegos de encaje son aquellos cuya función es encajar unas piezas dentro de otras, que usualmente se presentan perforadas con la misma figura, también existen algunos que cumplen doble función y se utilizan como rompecabezas. Estos pueden fabricarse con materiales de provecho, tales como cajas, cartones, gaveras de huevos, rollos de papel, chapas de refresco, discos compactos.

#### B. Importancia

Estimulación Temprana para Padres [7] nos explica que la importancia de los encajes y rompecabezas es que crecen con los niños. Mientras más piezas tengan, más dificultad se tendrá para armarlo y se desarrollará su inteligencia, concentración y su capacidad de resolver dificultades. Generalmente, se comienzan con encajes sencillos, como bloques grandes para encajar en un cubo para después pasar a encajes de una pieza, así cada vez pasando hasta los rompecabezas donde los niños comenzarán uniendo las piezas por las esquinas, luego por los bordes y, luego, se colocan las fichas del centro.

Los encajes son herramientas importantes en el aprendizaje educativo que proporciona el desarrollo de muchas habilidades mentales como lo son las habilidades cognitivas, resolución de problemas, desarrollo de la motricidad fina, coordinación de ojos y manos, social, autoestima.

#### C. Objetivos

Aeuno, usuario del blog Evaflex, [8] nos dice que el objetivo general es que los niños adquieran de forma gradual una autonomía que le permita valerse de manera adecuada e integral en sus habilidades corporales, intelectuales y socioemocionales a través de la exploración de las formas.

Por otra parte los objetivos específicos son que el niño distinga a través de la indagación la forma, el tamaño y el color de los objetos. También les enseña a distinguir las figuras geométricas básicas, gracias a este producto el infante podrá manifestar iniciativa para explorar su medio y entender su interés de conocer los elementos.

### IV. HABILIDADES MOTRICES

### A. Definición

El sitio web alicante [9] define habilidades motrices como el grado de competencia de un sujeto en concreto frente a una tarea determinada. Correr, saltar, lanzar, son habilidades motrices básicas. Dentro de la habilidades motrices básicas se encuentran: de locomoción, no-locomoción y de proyección-recepción. Suelen dividirse en Psicomotricidad Gruesa y Fina

### B. Motricidad Gruesa

Ardanaz [10] nos dice que la psicomotricidad gruesa es el control que se tiene sobre el propio cuerpo, especialmente los movimientos globales y amplios dirigidos a todo el cuerpo. Se refiere a aquellas acciones realizadas con la totalidad del cuerpo, coordinando desplazamientos y movimiento de las diferentes extremidades, equilibrio, y todos los sentidos. Cambiar, correr, rodar, saltar, girar, deportes, expresión corporal, entre otros están en esta categoría.

### C. Motricidad Fina

De igual forma Ardanaz [10] señala que la psicomotricidad fina se corresponde con las actividades que necesitan precisión y un mayor nivel de coordinación. Se refiere a movimientos realizados por una o varias partes del cuerpo. El niño inicia la psicomotricidad fina alrededor del año y medio, ya que implica un nivel de maduración y un aprendizaje previo.

## V. LIBRERÍAS, DOCUMENTACIÓN Y RECURSOS UTILIZADOS Y DEMÁS REQUERIMIENTOS

### A. Lenguaje de Programación

El lenguaje de Programación que se utilizó para la elaboración de este proyecto fue Python debido a la facilidad de comprensión y lectura en su sintaxis así como al ser un lenguaje interpretado no necesita un proceso de compilación, pero la principal razón es su facilidad para el análisis de casi cualquier tipo de datos, y la librería externa OpenCV para las imágenes y visión artificial; esta librería pasará a ser explicada en la siguiente parte de esta sección. Es importante recalcar que toda la información acerca de las librerías utilizadas que serán explicadas fue extraída de su documentación y/o sitios oficiales tanto de OpenCV [11] [12], como de PyQt [13] [14] y Numpy [15].

La versión de Python utilizada para la elaboración de este software fue Python 3.7.1.

### B. OpenCV

#### 1. Definición

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión de computadora y de aprendizaje automático de código abierto. OpenCV fue construido para proporcionar una infraestructura común para aplicaciones de visión artificial y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia BSD, OpenCV facilita que las empresas utilicen y modifiquen el código [11]. La versión utilizada para la elaboración del software fue la 3.4.3.18.

#### 2. Recursos Utilizados

- *cv2.VideoCapture(device)*: Constructor de la clase *VideCapture* que sirve para la captura de imágenes o video donde *device* hace la función de ID o índice de la cámara, si solo hay una se le deberá enviar 0.
- *cv2.VideoCapture.get(propId)*: Retorna el valor de una propiedad especificada.
- *cv2.VideoCapture.read()*: Graba, decodifica y devuelve el siguiente fotograma de video.
- *cv2.inRange(src, lowerb, upperb[, dst])*: Verifica si *src* esta entre los arreglos *lowerb* y *upperb*.
- *cv2.morphologyEx(src, op, kernel)*: Realiza transformaciones morfológicas
- *cv2.boundingRect(points)*: Calcula el rectángulo delimitador de arriba a la derecha de un conjunto de puntos.

### C. PyQt

#### 1. Definición

PyQt es un conjunto de enlaces Python v2 y v3 para el marco de la aplicación Qt de The Qt Company y se ejecuta en todas las plataformas compatibles con Qt, incluyendo Windows, OS X, Linux, iOS y Android. La versión utilizada es 5.9.2

#### 2. Recursos Utilizados

- *width(), height(), pos().x(), pos().y()*: Este conjunto de métodos que poseen todos los objetos de PyQt retornan el ancho, el alto, la posición en el x e y respectivamente.

### D. Numpy

## 1. Definición

NumPy es el paquete fundamental para la computación científica en Python. Es una biblioteca de Python que proporciona un objeto de matriz multidimensional, varios objetos derivados (como matrices y matrices enmascarados) y una variedad de rutinas para operaciones rápidas en matrices, incluyendo matemática, lógica, manipulación de formas, clasificación, selección, E/S. , transformadas de Fourier discretas, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria y mucho más. Usamos la versión 1.15.3.

## 2. Recursos Utilizados

- `numpy.ones(shape, dtype=None, order='C')`: Devuelve un nuevo conjunto de formas y tipos dados, llenos de unos. Shape es la forma que tendrá la matriz, dtype es el tipo de dato, order es el orden de las filas.
- `numpy.uint8`: Tipo de dato entero sin signo que varía que en un rango [0,255].
- `numpy.array(object)`: Crea una matriz, recibe object que debe tener una metodo que devuelve una matriz.

# VI. RED GLOVE

## A. Objetivo

Red Glove es un juego el cual se basa en los juegos didácticos de encaje, en los cuales como ya se vio consisten en poner piezas o bloques en lugares donde encajen con el objetivo de desarrollar la motricidad gruesa.

El objetivo principal de este software es incentivar la implementación de la tecnología, especialmente las Tecnologías de Información y Comunicación (TIC's) en ámbitos sociales como la Accesibilidad a software para discapacitados, para este caso en concreto el objetivo será el ser una herramienta auxiliar, es decir, de ayuda en el campo de la estimulación temprana en el desarrollo de las habilidades motrices.

## B. Interfaz Gráfica

Como se puede apreciar en la Figura 1 la interfaz gráfica es muy sencilla tiene 6 compartimientos; en forma de estrella, triángulo, pentágono, cuadrado, luna en cuarto menguante y un círculo; donde se debe “encajar” la pieza que aparece de modo aleatorio, en caso de la Figura 1 la pieza que apareció fue un pentágono, la parte más importante y funcional es la

mano con un dedo índice y unos círculos hace la función de puntero y hace funcionar todo el software, adicionalmente cuenta con una sección que hace de botón con finalidad de detener la ejecución sin haber ganado.



Figura 1: Interfaz Gráfica de Red Glove

La interfaz gráfica fue creada con PyQt5. En la Figura 2 se presenta el listado de los elementos de la Interfaz Gráfica; de b1 a b6 son los campos donde debe ser insertada la forma que en la interfaz es llamada mov, el widget que hace de cursor en el software es cs globos es un indicador de avance que va aumentando junto a score\_ui y surrender es el botón para terminar el programa sin haber ganado

Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
b1	QWidget
b2	QWidget
b3	QWidget
b4	QWidget
b5	QWidget
b6	QWidget
cs	QWidget
globos	QWidget
label	QLabel
mov	QWidget
score_ui	QLabel
surrender	QPushButton

Figura 2: Elementos de la Interfaz Gráfica

## C. Modo de Juego

El modo de jugar es simple, para esto se necesita tener puesto un guante de color rojo, una vez que ha cargado todo lo necesario cs empezará a moverse en la dirección en que movamos la mano con el guante, se deberá cerrar la mano para poder arrastrar mov a hacia la pieza señalada, y abrirla para soltarlo, cada cierto aumentará 100 puntos y globos

crecerá cuando este último llegue a cruzar toda la pantalla habrá ganado. En la Figura 3 se muestra el comportamiento del juego cuando se está “arrastrando”.

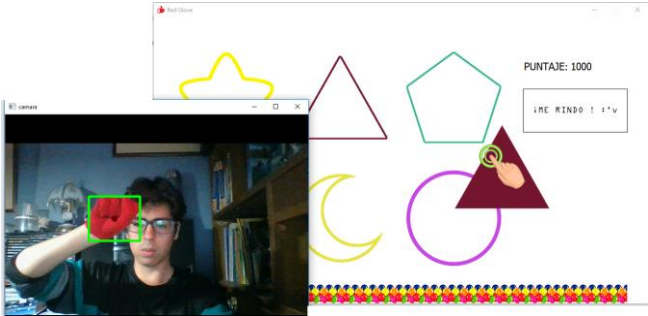


Figura 3: Modalidad de juego

VII. DESARROLLO E IMPLEMENTACIÓN

A. Versiones Anteriores

1. Versión 1

La versión 1 era una lector para archivos pdf que cambiaba de página con la mano. Esta versión solo funcionaba por consola y requería un movimiento de 30 píxeles a la derecha o izquierda para el cambio de página. En la Figura 4 se muestra su funcionamiento por consola.

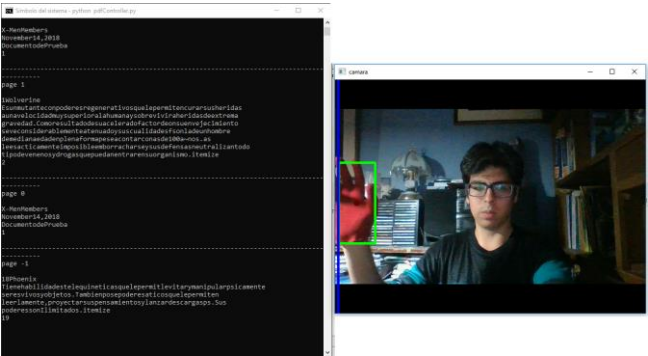


Figura 4: Uso de la versión 1

2. Versión 2

En la segunda versión se buscó un modo más interactivo por ello se realizó un juego en el que se podía seleccionar como personaje entre Dr. Strange y Scarlet Witch (personajes de The Avengers). Al seleccionar el personaje aparece otra ventana donde se puede movilizar con el pero no se puede hacer más que perder al tocar el fuego. Estas interfaces se pueden observar en la Figura 5 y 6

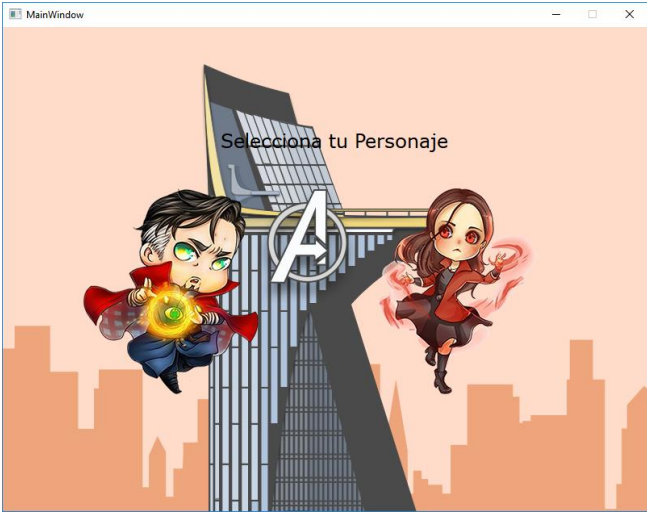


Figura 5: Interfaz de selección de personaje

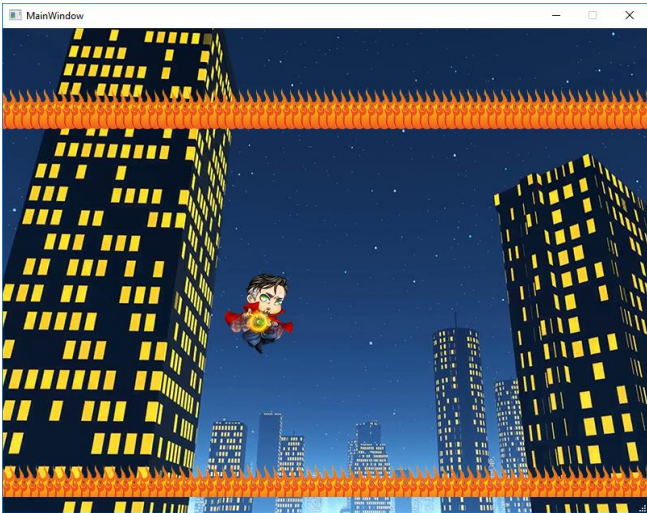
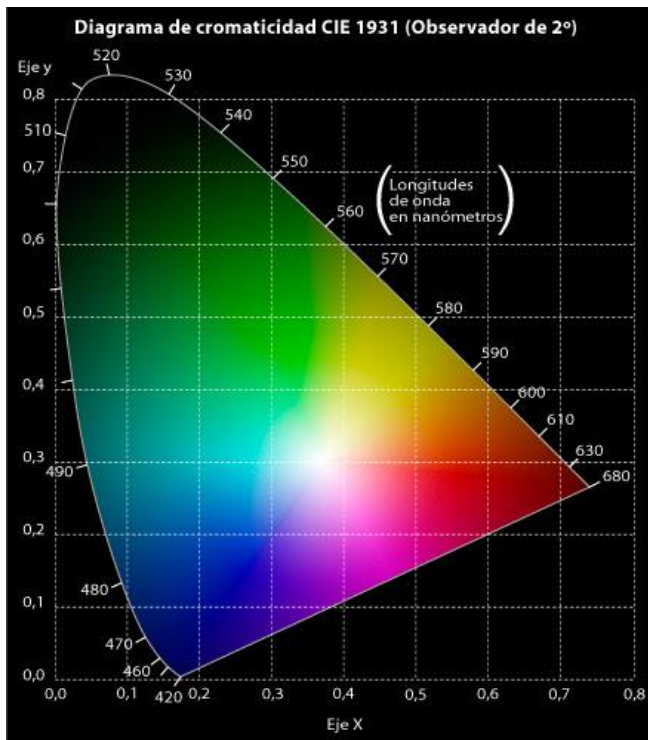


Figura 6: Interfaz jugable

B. Color Rojo

En este punto es importante justificar la elección de del color rojo como motor de todo el software en sí, Como se puede observar en la Figura 7, a diferencia de otros colores, el color rojo tiene menor variación en sus tonalidades y el área que ocupa en cada en la gráfica como se puede observar es la menor.



**Figura 7: Diagrama de Colores**

### C. Código Fuente

El código para la detección de colores fue sacado de el video [16] que fue puesto en el repositorio de GitHub [17]. El software fue desarrollado en el IDE PyCharm en su versión open source A continuación se pasará a presentar y explicar el código usado.

En la Figura 8 se observa todas las librerías necesarias para el software de la línea 1 a la 7

```

1 import cv2
2 import sys
3 import numpy as np
4 from PyQt5.QtWidgets import QApplication, QMainWindow, QMessageBox
5 from PyQt5 import uic
6 import random
7 import ctypes

```

**Figura 8: Librerías importadas**

En la Figura 9 se observa el proceso de ejecución del programa; las líneas 109 y 110 no son tan importantes ya que solo sirven para que el programa use el icono asignado en PyQt, las líneas 111 y 114 sirve para poder ejecutar correctamente el archivo ui creado en PyQt la línea 112 crea un objeto del clase Ventana y en la 113 se realiza el método empezar de la clase ventana que es la ejecución del juego en si

```

109 myappid = u'mycompany.myproduct.subproduct.version'
110 ctypes.windll.shell32.SetCurrentProcessExplicitAppUserModelID(myappid)
111 app = QApplication(sys.argv)
112 ventana = Ventana()
113 ventana.empezar()
114 app.exec_()

```

**Figura 9: Proceso de ejecución del programa**

En la Figura 10 se observa el constructor de la clase Ventana en el se crean atributos importantes como arr que guarda los campos de aceptación que vimos en la sección de Interfaz Gráfica de igual modo el img que guarda los nombres de los archivos de sus respuestas, también se seta los atributos de x e y con las posiciones un campo de aceptación aleatorio y se cambia la imagen de mov a su respuesta; se inicia la puntuación en 0.

```

9 class Ventana(QMainWindow):
10     def __init__(self):
11         QMainWindow.__init__(self)
12         self.camara = True
13         uic.loadUi("ui.ui", self)
14         self.show()
15         self.arr = [self.b1, self.b2, self.b5, self.b3, self.b4, self.b6]
16         self.img = ['starform.png', 'triangleform.png', 'pentagonform.png']
17         a = random.randint(0,5)
18         self.mov.setStyleSheet("background: url('"+self.img[a]+"');")
19         self.x = self.arr[a].pos().x()
20         self.y = self.arr[a].pos().y()
21         self.score = 0
22         string = str(self.score)
23         self.score_ui.setText(string)
24         self.run = True

```

**Figura 10: Constructor de la clase Ventana**

En la Figura 11 se observa en las líneas 27 y 28 la preparación para la toma de imágenes, widthcam y heightcam sirven para convertir las coordenadas de la cámara a las de la interfaz en proporción. punto x e y guardaran las posiciones captadas por la cámara que servirá junto con distmin para conseguir una mejor precisión, catch ayudará a saber si se ha cogido mov , posx0 e y0 almacenarán la posición original de mov y los surrenders los datos del botón para rendirse.

```

26 def empezar(self):
27     cam = cv2.VideoCapture(0)
28     kernel = np.ones((5, 5), np.uint8)
29     widthcam = cam.get(3)
30     heightcam = cam.get(4)
31     puntox = 0
32     puntoy = 0
33     distmin = 3
34     catch = False
35     area = 0
36     posx0 = self.mov.pos().x()
37     posy0 = self.mov.pos().y()
38     surrenderox = self.surrender.pos().x()
39     surrenderoxw = surrenderox + self.surrender.width()
40     surrendery = self.surrender.pos().y()
41     surrenderyh = surrendery + self.surrender.height()

```

**Figura 11: Inicio del método empezar**

En la Figura 12 nos muestra de las líneas de la 43 a la 48 la captura de los datos relevantes como lo son la posición ancho y altura del guante en cada iteración que será almacenado en las variables x, y, w, h respectivamente, de la línea 49 a la 53 se dibuja en la cámara , en caso de mostrarse, un rectángulo bordeando el guante y un punto en el centro.



```

42 while (self.run):
43     ret, frame = cam.read()
44     rangomax = np.array([220, 60, 255]) #30 30 255
45     rangomin = np.array([25, 20, 100]) #10 10 80
46     mascara = cv2.inRange(frame, rangomin, rangomax)
47     opening = cv2.morphologyEx(mascara, cv2.MORPH_OPEN, kernel)
48     x, y, w, h = cv2.boundingRect(opening)
49     cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 4)
50     mediox = round(x + w / 2)
51     medioy = round(y + h / 2)
52     cv2.circle(frame, (mediox, medioy), 6, (0, 0, 100), -1)

```

**Figura 12:** Segunda parte de empezar

En la Figura 13 observamos que con los datos obtenidos vemos que la distancia entre en el punto anterior y el actual varía en 3, que es la distmin, o más para mejorar la precisión, si eso se cumple se convierten las coordenadas de la cámara a la interfaz y si el área anterior disminuye proporcionalmente y no se ha agarrado la pieza se evalúa si la posición obtenida está dentro de la posición inicial de mov catch cambia a true, si no si está dentro de rendirse se le pregunta al usuario si desea rendirse en ese caso se termina el bucle.

```

53 if abs(mediox - puntiox) > distmin and abs(medioy - puntioy) >= distmin:
54     xframe = self.frameSize().width() - ((self.frameSize().width() * mediox) / widthcam)
55     yframe = (800 * medioy) / (heightcam) - 100 # self.frameSize().height()
56     puntiox = mediox
57     puntioy = medioy
58     if h*w - area < -area*1/5 and not catch:
59         if posxo+25 < xframe < posxo+175 and posyo + 25 < yframe < posyo + 175 and not catch:
60             catch=True
61
62     elif surrendrx < xframe < surrendrxw and surrendry < yframe < surrendryh:
63         resultado = QMessageBox.question(self, "¡Rendirse!", "¿Estas seguro de querer rendir")
64         if resultado == QMessageBox.Yes:
65             break

```

**Figura 13:** Tercera parte de empezar

En la Figura 14 vemos que en caso no se presente la disminución del área puede aumentar y si esto pasa catch cambia a false es decir se debe soltar mov, si mov es soltado en la posición correcta score y globos deben de aumentar y resetearse los valores de mov; si despues de esto globos alcanzó el ancho de la ventana se ha ganado ek juego y se le pregunta al usuario si quiere volver a jugar en ese caso score y globos se inician en 0 de lo contrario se termina el bucle.

```

67 elif area - h * w < -area*1/2 and catch:
68     catch=False
69     if self.x < xframe < self.x+200 and self.y - 100 < yframe < self.y + 100:
70         self.score = self.score+100
71         string = str(self.score)
72         self.score_ui.setText(string)
73         a = random.randint(0, 5)
74         self.mov.setStyleSheet("background: url('\" + self.img[a] + \");")
75         self.x = self.arr[a].pos().x()
76         self.y = self.arr[a].pos().y()
77         self.globos.setMaximumSize(self.score, 40)
78         self.globos.setMinimumSize(self.score, 40)
79         if self.width() <= self.globos.width():
80             resultado = QMessageBox.question(self, "¡GANASTE!", "¿Deseas volver")
81             if resultado == QMessageBox.Yes:
82                 self.score = 0
83                 string = str(self.score)
84                 self.score_ui.setText(string)
85                 self.globos.setMaximumSize(0, 0)
86                 self.globos.setMinimumSize(0, 0)
87             else:
88                 break
89

```

**Figura 14:** Cuarta parte de empezar

Si no paso los casos de las Figuras 13 y 14, la 15 nos dice que se debe mover cs a las coordenadas de interfaz, si se agarrado L. E. Caiza Saquipulla and F. R. Campoverde Balla, "Automatización de Actividades Lúdicas Mediante el Uso de Interfaces Naturales de Usuario, para el

mov; es decir, catch es igual a true, mov tambien es cambiado de posición. La línea 100 está comentada debido a que esa muestra la captura de la cámara sin embargo su ejecución no afecta el programa pero es de ayuda para calcular los rangos máximos y mínimos.

```

90 else:
91     self.cs.move(xframe, yframe)
92     area = h * w
93     if (catch):
94         self.mov.move(xframe-50, yframe-50)
95     else:
96         self.mov.move(posxo, posyo)
97
98
99
100 #cv2.imshow('camara', frame)

```

**Figura 15:** Final de empezar

## VIII. CONCLUSIÓN

En conclusión, Red Glove es una herramienta que podría ser útil en ser útil para el desarrollo de habilidades motrices tanto gruesa como fina; sin embargo, seria de mas ayuda para la gruesa debido a que son movimientos no tan finos los que se realizan en el software, el problema en sí que tiene este software es que depende de la cantidad y tipo de luz del ambiente, ya que eso cambiaría los rangos para la percepción del color lo que causaría errores con objetos de color naranja y amarillo e incluso con la piel.

## IX. REFERENCIAS

- [1] L. Jácome Amores, L. J. Amores, and J. Jadán-Guerrero, "TEVI: Teclado virtual como herramienta de asistencia en la comunicación y el aprendizaje de personas con problemas del lenguaje vinculados a la discapacidad motriz," *Káñina*, vol. 40, no. 4, p. 105, 2017.
- [2] B. Muro Haro, P. Santana Mancilla, and M. García Ruiz, "Uso de interfaces tangibles en la enseñanza de lectura a niños con síndrome de Down," *El Hombre y la Máquina*, no. 39, pp. 19–25, 2012.
- [3] E. Rivera Capa, L. Quispe de la Cruz, and C. Montalvo Yarnold, "Realidad Aumentada e Inteligencias Múltiples en el Aprendizaje de Matemáticas," *Concurso de Proyectos Feria Tecnológica IEEE INTERCON*, 2011.
- [4] F. Días, M. Vizcanio, C. Banchoff, E. Lanfranco, and J. Bogado, "Implementacion de un programa (PyMoHa) para controlar el mouse mediante el uso de un pulsador para personas con discapacidades motrices graves," in *XIII Workshop de Investigadores en Ciencias de la Computación*, 2011.
- [5]

Desarrollo de la Motricidad Gruesa en los Niños con Discapacidad Intelectua de Primero a Decimo Año en Educación General Básica de la Unidad Educativa

- Especializada ‘Carlos Gabay,’” Tesis de Grado, Universidad Nacional de Chimborazo, 2016.
- [6] “Juegos de Encajes y Ensates,” *RECURSOS DEL APRENDIZAJE*, 22-Mar-2012. [Online]. Available: <https://recursosdelaprendizaje.wordpress.com/recursos-didacticos/juegos-de-encajes-y-ensates/>. [Accessed: 05-Dec-2018].
- [7] “Importancia de los encajes y rompecabezas | Revista Online Estimulación Temprana Para Padres,” *Revista Online ETPP - Estimulación Temprana Para Padres*, 19-May-2017. [Online]. Available: <http://www.estimulaciontempranapapadres.com/articulos-recomendados/importancia-los-encajes-rompecabezas/>. [Accessed: 05-Dec-2018].
- [8] E. E. Aeuno and V. T. mi Perfil, “Juegos para aprender: Juego de encajes.” [Online]. Available: <http://evaflex.blogspot.com/2013/08/juegos-para-aprender-juego-de-encajes.html>. [Accessed: 05-Dec-2018].
- [9] -alicante.com, “Habilidades motrices [Motor skills] (Educación física),” Aug. 2015.
- [10] T. Ardanaz García, “Psicomotricidad y educación infantil,” *Impreso en España*, 1994.
- [11] “OpenCV library.” [Online]. Available: <https://opencv.org/>. [Accessed: 06-Dec-2018].
- [12] “OpenCV documentation index.” [Online]. Available: <https://docs.opencv.org>. [Accessed: 06-Dec-2018].
- [13] “Riverbank | Software | PyQt | What is PyQt?” [Online]. Available: <https://riverbankcomputing.com/software/pyqt>. [Accessed: 06-Dec-2018].
- [14] “PyQt5 Reference Guide — PyQt 5.11.1 Reference Guide.” [Online]. Available: <http://pyqt.sourceforge.net/Docs/PyQt5/>. [Accessed: 06-Dec-2018].
- [15] “Numpy and Scipy Documentation — Numpy and Scipy documentation.” [Online]. Available: <https://docs.scipy.org/doc/>. [Accessed: 06-Dec-2018].
- [16] *Detección de objetos con Python y OpenCV*. 2018.
- [17] JacksonDuvan, “JacksonDuvan/OpenCV-y-Python-,” *GitHub*. [Online]. Available: <https://github.com/JacksonDuvan/OpenCV-y-Python->. [Accessed: 06-Dec-2018].