Finger Identifier

+ wCam: int
+ hCam: int
+ camera: cv2.VideoCapture
+ pTime: float
+ cTime: float
+ fps: float
+ detector: handDetector
+ success: bool
+ frame: numpy.ndarray
+ lmList: list
+ key: int

+ indexFinger(lmList: list): int
+ middleFinger(lmList: list): int
+ ringFinger(lmList: list): int
+ pinkyFinger(lmList: list): int
+ thumbFinger(lmList: list): int
+ allFingers(lmList: list): int

FingerMovement

+ wCam: int
+ hCam: int
+ camera: cv2.VideoCapture
+ pTime: float
+ cTime: float
+ fps: float
+ detector: handDetector
+ success: bool
+ frame: numpy.ndarray
+ lmList: list
+ fingerComb: int
+ key: int

+ gameMovement(cameraVal: int): str

HandDirection

+ wCam: int
+ hCam: int
+ camera: cv2.VideoCapture
+ pTime: float
+ cTime: float
+ fps: float
+ detector: handDetector
+ success: bool
+ frame: numpy.ndarray
+ lmList: list
+ key: int

+ northHand(lmList: list): int
+ eastHand(lmList: list): int
+ westHand(lmList: list): int
+ southHand(lmList: list): int

Static Gestures Class

+ wCam: int
+ hCam: int
+ camera: cv2.VideoCapture
+ pTime: float
+ cTime: float
+ fps: float
+ detector: handDetector
+ success: bool
+ frame: numpy.ndarray
+ lmList: list
+ fingerComb: int
+ key: int

+ rock(lmList: list): int
+ gnarly(lmList: list): int
+ peace(lmList: list): int
+ fist(lmList: list): int
+ loser(lmList: list): int

HandTrackingModule

+ main()

handDetector

+ mode: bool
+ maxHands: int
+ modelComplex: int
+ detectionCon: float
+ trackCon: float
+ mpHands: module
+ hands: mediapipe.python.solutions.hands.Hands
+ mpDraw: module
+ results: type

+ handDetector(mode: bool, maxHands: int,
 modelComplex: int, detectionCon: float,
 trackCon: float)
+ findHands(img: numpy.ndarray, draw: bool):
 numpy.ndarray
+ findPosition(img: numpy.ndarray, handNo: int,
 draw: bool): list
+ fingerCombination(lmList: list): int

uses 1 ... 1

uses 1 ... 1

uses 1 ... 1

uses 1 ... 1

uses 1 ... 1