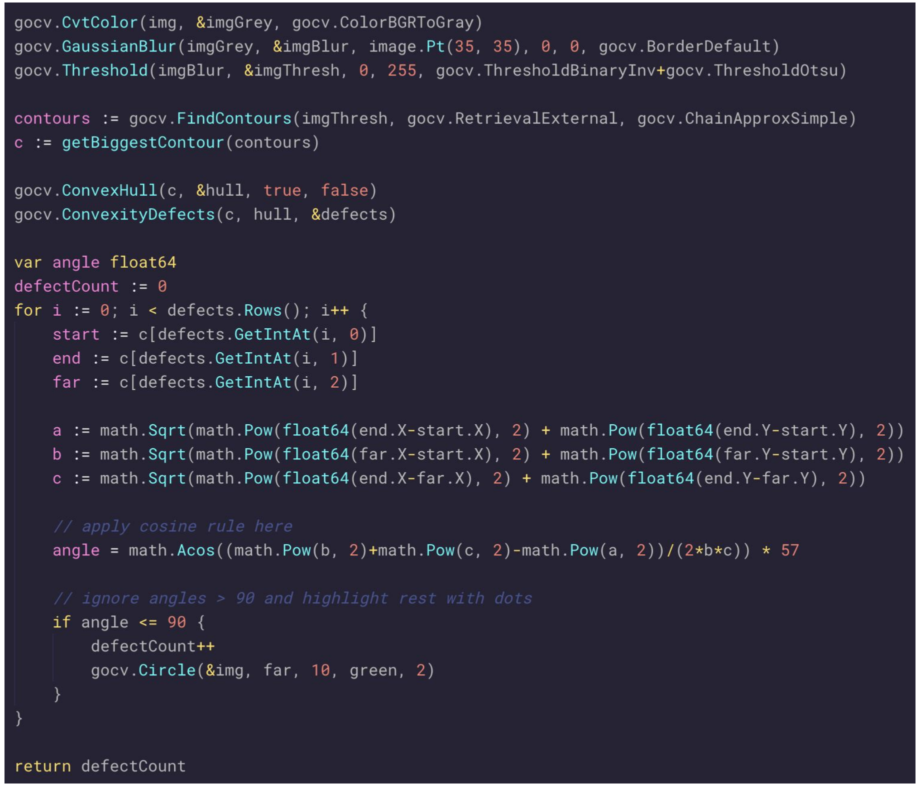
Rysowanie w aplikacji, śledząc ruch markera przy pomocy systemu wizyjnego

Grupa projektowa: Wojciech Niedbała, Albert Millert, Michał Mirończuk, Patryk Romaniak

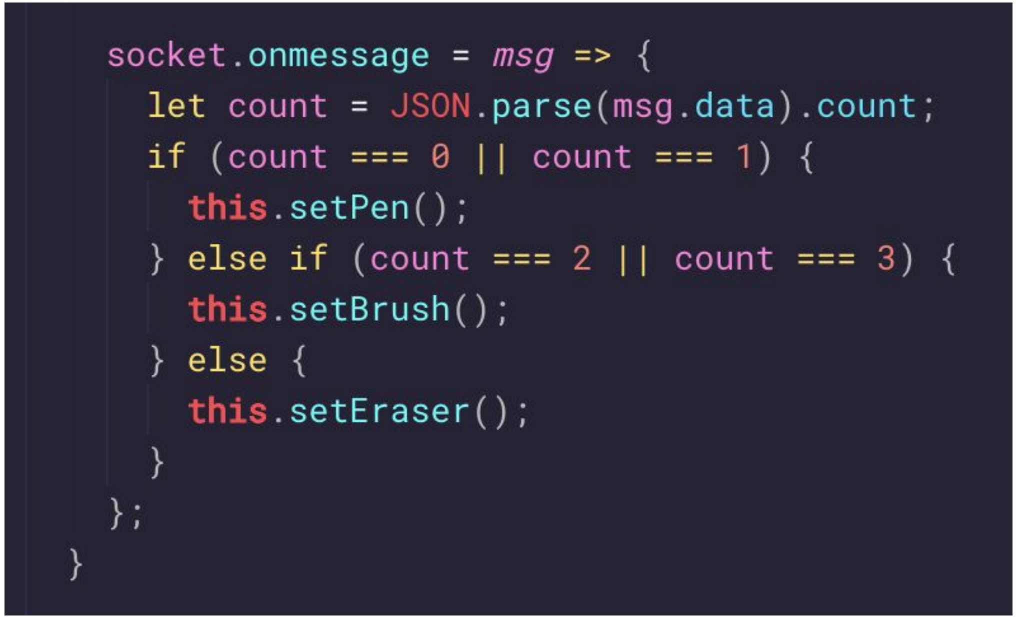
Część V

Gesty



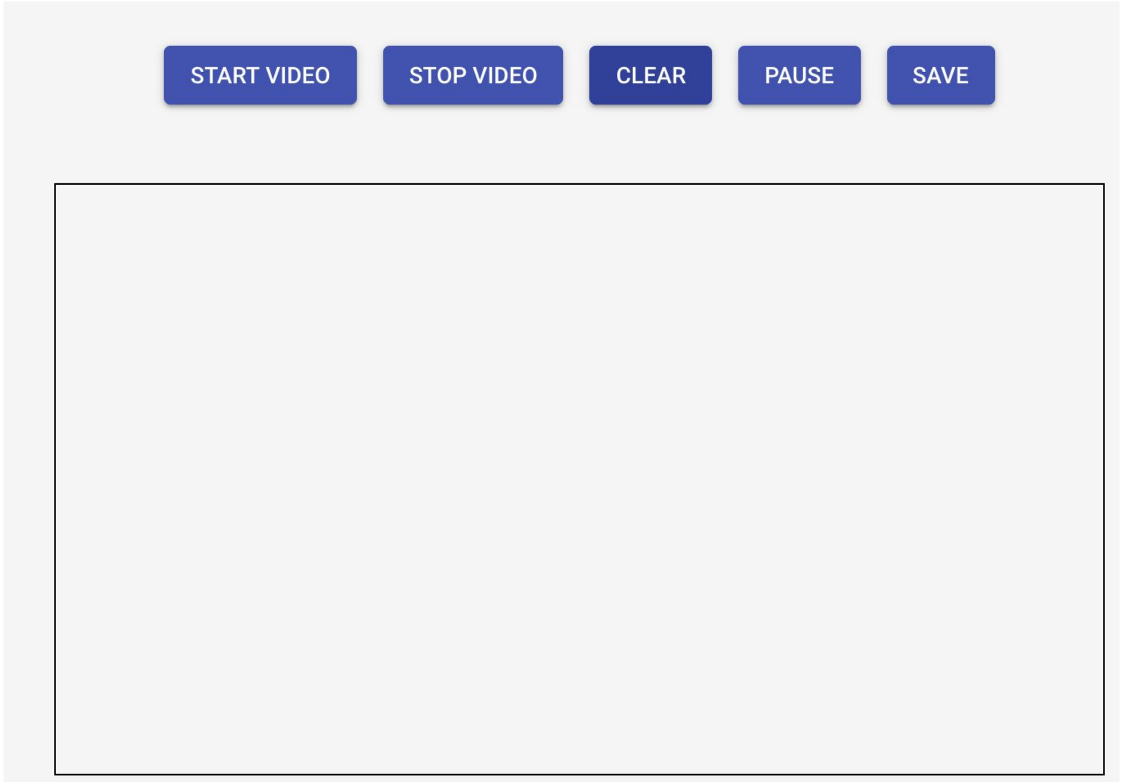
```go
gocv.CvtColor(img, &imgGrey, gocv.ColorBGRToGray)
gocv.GaussianBlur(imgGrey, &imgBlur, image.Pt(35, 35), 0, 0, gocv.BorderDefault)
gocv.Threshold(imgBlur, &imgThresh, 0, 255, gocv.ThresholdBinaryInv+gocv.ThresholdOtsu)

contours := gocv.FindContours(imgThresh, gocv.RetrievalExternal, gocv.ChainApproxSimple)
c := getBiggestContour(contours)

gocv.ConvexHull(c, &hull, true, false)
gocv.ConvexityDefects(c, hull, &defects)

var angle float64
defectCount := 0
for i := 0; i < defects.Rows(); i++ {
    start := c[defects.GetIntAt(i, 0)]
    end := c[defects.GetIntAt(i, 1)]
    far := c[defects.GetIntAt(i, 2)]

    a := math.Sqrt(math.Pow(float64(end.X-start.X), 2) + math.Pow(float64(end.Y-start.Y), 2))
    b := math.Sqrt(math.Pow(float64(far.X-start.X), 2) + math.Pow(float64(far.Y-start.Y), 2))
    c := math.Sqrt(math.Pow(float64(end.X-far.X), 2) + math.Pow(float64(end.Y-far.Y), 2))

    // apply cosine rule here
    angle = math.Acos((math.Pow(b, 2)+math.Pow(c, 2)-math.Pow(a, 2))/(2*b*c)) * 57

    // ignore angles > 90 and highlight rest with dots
    if angle <= 90 {
        defectCount++
        gocv.Circle(&img, far, 10, green, 2)
    }
}

return defectCount
```

Gesty



```
    socket.onmessage = msg => {
      let count = JSON.parse(msg.data).count;
      if (count === 0 || count === 1) {
        this.setPen();
      } else if (count === 2 || count === 3) {
        this.setBrush();
      } else {
        this.setEraser();
      }
    };
  }
```

Pauzowanie i czyszczenie


START VIDEO
STOP VIDEO
CLEAR
PAUSE
SAVE