

# Computerpraktikum

---

## Protokoll

Thema: Raspberry Iriserkennung

Datum: 11.03 - 27.05.

Name: Mona Angerer, Julian Jaklitsch, David Thaller

Gruppe 2

Klasse: 2BHITM

Seite: 1 / 5

## Verlauf

---

- |  |            |
|--|------------|
| 1. Planung                                       | 25.02.2021 |
| 2. Anfang des Projektes, Einteilung              | 11.03.2021 |
| 3. Erweiterung und Verbesserung                  | 25.03.2021 |
| 4. Kamera wurde aktiviert                        | 22.04.2021 |
| 5. beschädigte Ordnerstruktur                    | 29.04.2021 |
| 6. neuer und besserer Raspberry                  | 06.05.2021 |
| 7. Recherche zu ChartJS                          | 20.05.2021 |
| 8. Iriserkennung erfolgreich mit neuem Raspberry | 27.05.2021 |

## 1. Planung

---

### Iriserkennung

---

#### Ziele

- Auge/Iris wird erkannt
- Unterschied erkennen zwischen Auge und Objekt
- schönes Design mit js Elementen
- strukturierte Programmierung

#### Nichtziele

- merken von Augenpaaren

#### Präzise Themenbeschreibung

Zuerst wird programmiert und eine Schaltung erstellt.

Ein wichtiges Bauteil ist die Kamera, welche für das Erkennen des Auges zuständig ist.

Zusätzlich tragen Soft und Hardware einen Teil dazu bei.

# Computerpraktikum

## Protokoll

Thema: Raspberry Iriserkennung

Datum: 11.03 - 27.05.

Name: Mona Angerer, Julian Jaklitsch, David Thaller

Gruppe 2

Klasse: 2BHITM

Seite: 2 / 5

### Verwendete Technologien

#### SW

- [OpenCV](#)
- Java Framework: highChart oder chartJS
- [Docker](#)

#### HW

- [Raspberry Pi Webcam](#)
- [Raspberry Pi IR Cam](#)
- Raspberry Pi Noir Camera Module V2

### Teamzusammensetzung (Projektleiter muss definiert werden)

**Projektleiter:** David Thaller

**Teammitglieder:** Mona Angerer, Julian Jaklitsch

## 2. Anfang des Projektes, Einteilung

- Einlesen ins Thema Raspberry Pi Cam(<https://electreeks.de/raspberry-pi-kamera-installieren-anschliessen/>) und wie man ein Image erstellt
- SD-Karte in Laptop stecken und Image draufladen
- Raspberry aus der Verpackung holen
- Maus, Tastatur an Raspberry anstecken und mit Bildschirm verbinden
- Image einstecken und wenn vorhanden Lan-kabel anschließen
- Netzkabel an Steckdose anschließen und mit Raspberry verbinden
- Raspberry starten
- In Terminal ein Update laden mit der Zeile: "SUDO APT-GET UPDATE && SUDO APT-GET UPGRADE" und anschließend die Raspberrykonfiguration öffnen mit "SUDO RASPI-CONFIG" um festzustellen welche Sprache und Tastatur ausgewählt ist

# Computerpraktikum

## Protokoll

Thema: Raspberry Iriserkennung

Datum: 11.03 - 27.05.

Name: Mona Angerer, Julian Jaklitsch, David Thaller

Gruppe 2

Klasse: 2BHITM

Seite: 3 / 5

### 3. Erweiterungen und Verbesserung

- Damit wir in der nächsten Stunde erfolgreich die Kamera anschließen haben wir uns in das Thema noch mal neu eingelesen da wir noch nicht wussten wie das Terminal funktioniert
- Die restliche Stunden haben wir mit der suche verbracht

### 4. Kamera wurde aktiviert

- Kamera auspacken und anschließen und Raspberry rebooten
- Kamera ausgewählt
- neu gebootet
- im Terminal überprüft, ob die Kamera funktioniert, mit dem Befehl "VCENCMD\_CAMERA"
- -F" die Kamera gestartet und mit "RASPISTILL -T -O IMGNAME.jpg" ein Bild gemacht, mit entsprechender Zeitverzögerung -> Es wurde erfolgreich gesteckt und mithilfe des Terminal wurde es wurde mit dem Befehl "RASPISTILL die Kamera aktiviert. Zusätzlich wurde ein Foto geschossen.

### 5. Beschädigte Ordnerstruktur

- Am Anfang der Stunde haben wir uns in die Erkennung der Iris eingelesen, um es später auszuführen
- Nach der Öffnung des Terminals haben wir die ersten Befehle eingeben und es kamen immer Fehlermeldungen
- Wir fragten einen Lehrer was das Problem ist
- Das Problem war, dass die Ordnerstruktur des Images nicht mehr stimmte
- Das Image musste gelöscht werden und nochmal installiert werden

# Computerpraktikum

## Protokoll

Thema: Raspberry Iriserkennung

Datum: 11.03 - 27.05.

Name: Mona Angerer, Julian Jaklitsch, David Thaller

Gruppe 2

Klasse: 2BHITM

Seite: 4 / 5

- Somit standen wir wieder ganz am Anfang

## Neuer und besserer Raspberry

- Wir haben uns einen Raspberry PI 4 und eine neue, schnellere SD Karte zugelegt.
- Der Raspberry musste zusammengebaut werden, die Kamera angeschlossen und das Betriebssystem musste erneut installiert werden.

## Rechere zu ChartJS

- Was ist ChartJS? -> eine einfache und flexible JavaScript-Library, mit der man einfach Diagramme erstellen kann
- Wir haben ein Chart erstellt, jedoch mit random geneierten Werten und noch nicht mit denen aus unserer Gesichtserkennung

## Neuer und besserer Raspberry

- Wir haben uns einen Raspberry PI 4 und eine neue, schnellere SD Karte zugelegt.
- Der Raspberry musste zusammengebaut werden, die Kamera angeschlossen und das Betriebssystem musste erneut installiert werden.

## Iriserkennung erfolgreich mit neuem Rapsberry

- Phyton installiert
- file "camtest.py" erstellt und erfolgreich ausgeführt
- file "iris.py" erstellt
- Gesichtserkennung ausgearbeitet (wenn ein Gesicht erkannt wird, wird ein Rahmen um das Gesicht gezeigt)
- mittels eyecascades auch Augenerkennung programmiert (Kreise um beide Augen)

# Computerpraktikum

## Protokoll

Thema: Raspberry Iriserkennung

Datum: 11.03 - 27.05.

Name: Mona Angerer, Julian Jaklitsch, David Thaller

Gruppe 2

Klasse: 2BHITM

Seite: 5 / 5

```
#file „iris.py“
import io
import picamera
import cv2
import numpy

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_alt.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
while True:
    stream = io.BytesIO()
    with picamera.PiCamera() as camera:
        camera.resolution = (640, 480)
        camera.capture(stream, format='jpeg')
    buff = numpy.frombuffer(stream.getvalue(), dtype=numpy.uint8)

    image = cv2.imdecode(buff, 1)
    image = cv2.rotate(image, cv2.ROTATE_90_CLOCKWISE)
    image = cv2.rotate(image, cv2.ROTATE_90_CLOCKWISE)
    faces = face_cascade.detectMultiScale(image, 1.1, 5)
    eyes = eye_cascade.detectMultiScale(image, 1.1, 5)

    for (x,y,w,h) in faces:
        cv2.rectangle(image,(x,y),(x+w,y+h),(255,255,0),2)

    for (x,y,w,h) in eyes:
        cv2.circle(image,(x+int(w/2),y+int(w/2)),20,(255,255,255),2)

    cv2.imshow("show", image)
    if cv2.waitKey(100) & 0xFF == ord('x'):
        break
```