

# Schere-Stein-Papier mit KI-Bildererkennung

## Planung

### Aufgaben

- 2 Modell mit Teachable-Machine trainieren / Info-Text dazu verfassen (**Jonas, Nikolas**)
- 2 GitHub (**Jan-Niklas, Maxim**)
  - Account, Repository, Projekt-Webseite einrichten
  - Ergebnisse aus den anderen Gruppen sammeln und committen
- 2 HTML (GUI) programmieren (**Arne, Joel**)
  - Um die Wahl des Computers zu veranschaulichen könnte man die Hand/Gesten-Bilder aus dem Wikipedia-Artikel nehmen:  
[https://de.wikipedia.org/wiki/Schere,\\_Stein,\\_Papier](https://de.wikipedia.org/wiki/Schere,_Stein,_Papier)
- JavaScript (Logik) programmieren (Funktionsbeschreibungen s.u.)
  - 2 initModell(), loop() (**Dominik, Julian**)
  - 2 neuesSpiel(), neueRunde() (**Alimsah**)
  - 2 spielerZug(), computerZug() (**Fabian, Andre**)
  - 2 updateSpielstand(), updateGUI() (**Michael**)

**Zwischen einigen Aufgaben sind Absprachen nötig. Diese bitte, falls notwendig, eigeninitiativ durchführen!**

# JavaScript-Planung

## Globale (gemeinsame) Spiel-Variablen

(Innerhalb der einzelnen Funktionen können ggf. weitere lokale Variablen mit var deklariert werden.)

- computerWahl (0: Stein, 1: Papier, 2: Schere; Zufallsgenerator)
- spielerWahl (0: Stein, 1: Papier, 2: Schere; KI-Bilderkennung)
- computerZaehler (Anzahl gewonnener Runden des Computers)
- spielerZaehler (Anzahl gewonnener Runden des Spielers)
- runde (Nummer der aktuellen Runde)
- gewonnen (0: noch niemand gewonnen, 1: Computer gew., 2: Spieler gew.)

## Teachable-Machine-Modell-Variablen

- model
- webcam

## Funktionen

(if/else, switch/case etc. gibt es auch in JavaScript, ggf. recherchieren!!!)

- initModell() und loop()
  - Modell laden, Webcam starten und anzeigen, loop-Funktion starten
  - Siehe: <https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image>
- neuesSpiel()
  - wird auf Knopfdruck ausgelöst
  - alle Spiel-Variablen auf sinnvolle Anfangswerte zurücksetzen
  - updateGUI() aufrufen
- neueRunde()
  - wird auf Knopfdruck ausgelöst
  - Ching-Chang-Chong-Countdown
    - Mittels sleep-Funktion, siehe: <https://www.heise.de/tipps-tricks/JavaScript-Sleep-und-setTimeout-4060840.html>
  - Aufrufe der Methoden spielerZug(), computerZug(), updateSpielstand(), updateGUI()
- spielerZug()
  - Bild von der KI erkennen lassen und je nach zurückgegebenen Wahrscheinlichkeiten spielerWahl auf 0, 1 oder 2 setzen
  - Siehe: <https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image>
- computerZug()
  - Zufallszahl (0, 1 oder 2) erzeugen und computerWahl zuweisen.
  - Vordefinierte JavaScript-Funktionen Math.random und Math.floor recherchieren und verwenden!!!
- updateSpielstand()
  - computerWahl und spielerWahl vergleichen und feststellen, wer die Runde gewonnen hat
  - je nachdem computerZaehler oder spielerZaehler hochzählen
  - feststellen, ob das Gesamtspiel gewonnen wurde und entsprechend gewonnen-Variable setzen
- updateGUI()
  - Spiel-Variablen auslesen und dementsprechend alle relevanten HTML-Elemente / Texte updaten