

# Prezentacja sterowana gestami

*Dokumentacja Projektowa*

Autor:

**Filip Kula, Wiktor Mazur**

Przedmiot:

*Interakcja Człowiek - Komputer*

# 1 Wprowadzenie

Celem systemu jest zapewnienie interaktywnego sposobu zarządzania prezentacją bez konieczności użycia myszy lub klawiatury. Dzięki zastosowaniu technologii rozpoznawania gestów użytkownik może płynnie sterować slajdami i wykonywać na nich adnotacje.

## 2 Technologie

System został zbudowany w oparciu o następujące technologie:

- Next.js - framework do budowy aplikacji webowych,
- MediaPipe Hands - biblioteka do rozpoznawania gestów dłoni,
- Kontekst Canvas API - umożliwia rysowanie na slajdach,
- WebSocket - umożliwia komunikację w czasie rzeczywistym,
- Tailwind CSS - zapewnia estetyczny interfejs użytkownika.

## 3 Funkcjonalności

### 3.1 Zarządzanie slajdami

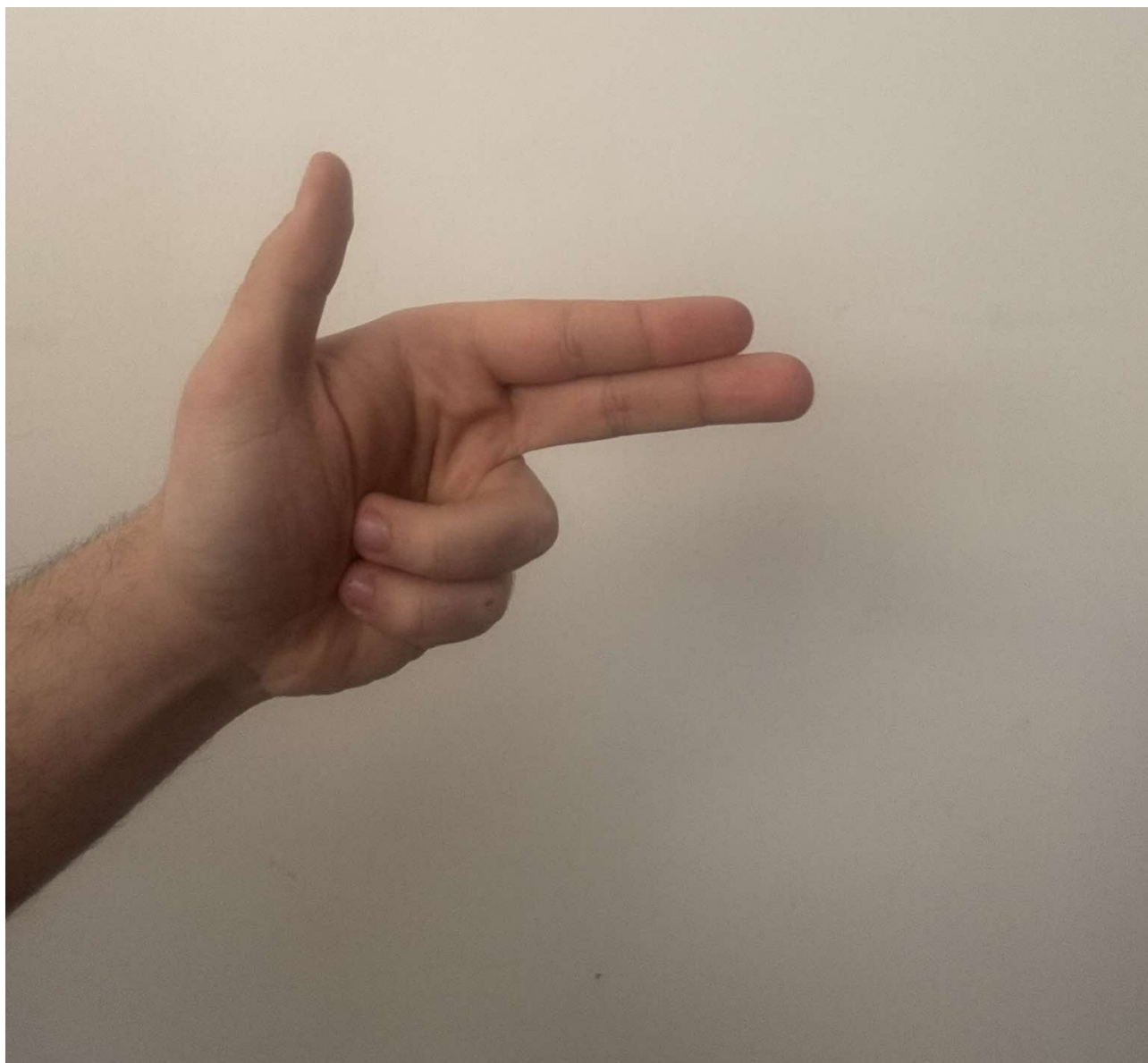
System umożliwia:

- Dodawanie nowych slajdów,
- Usuwanie istniejących slajdów,
- Nawigację między slajdami za pomocą gestów dłoni.

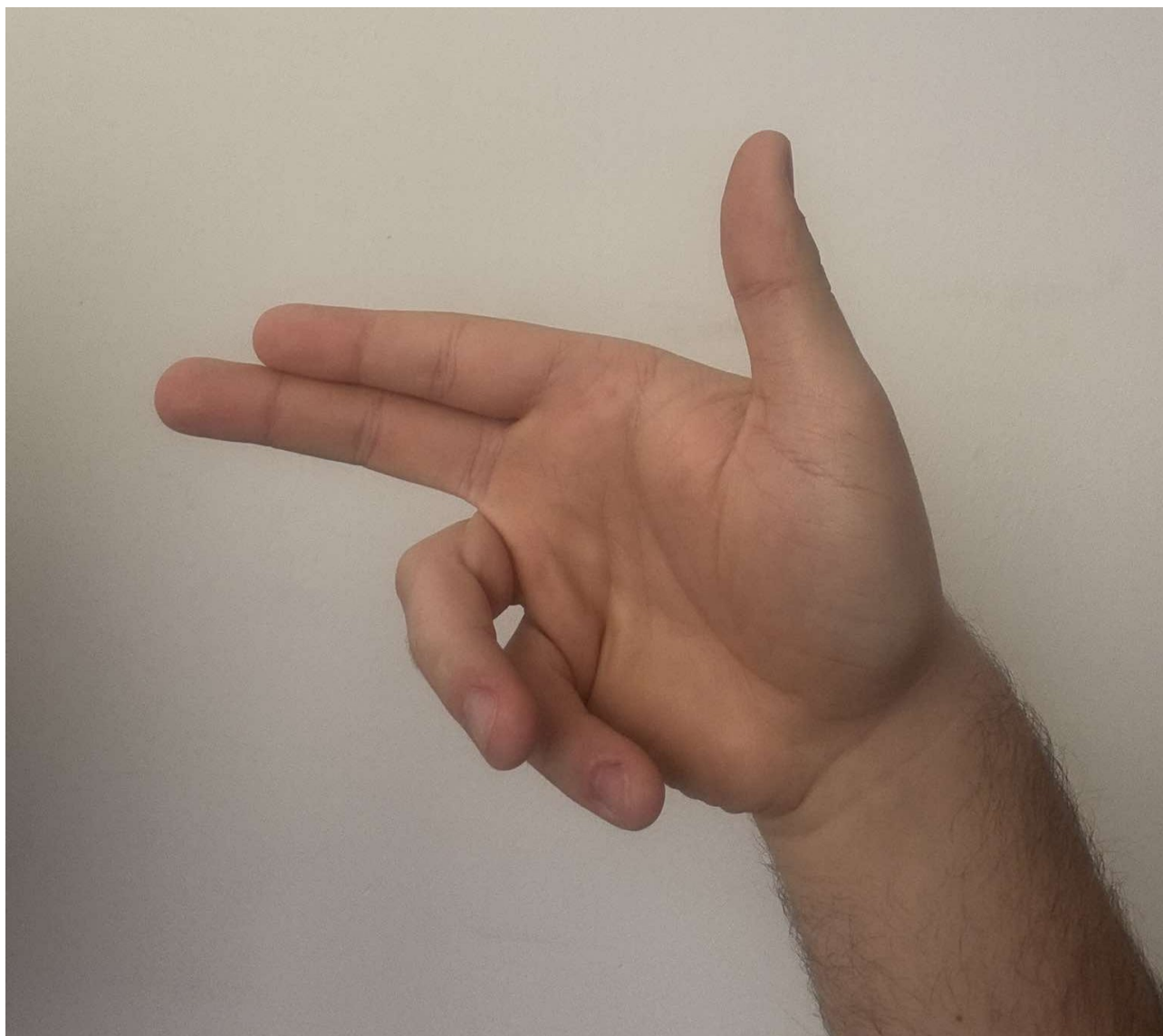
### 3.2 Obsługa gestów

Do sterowania prezentacją wykorzystano następujące gesty:

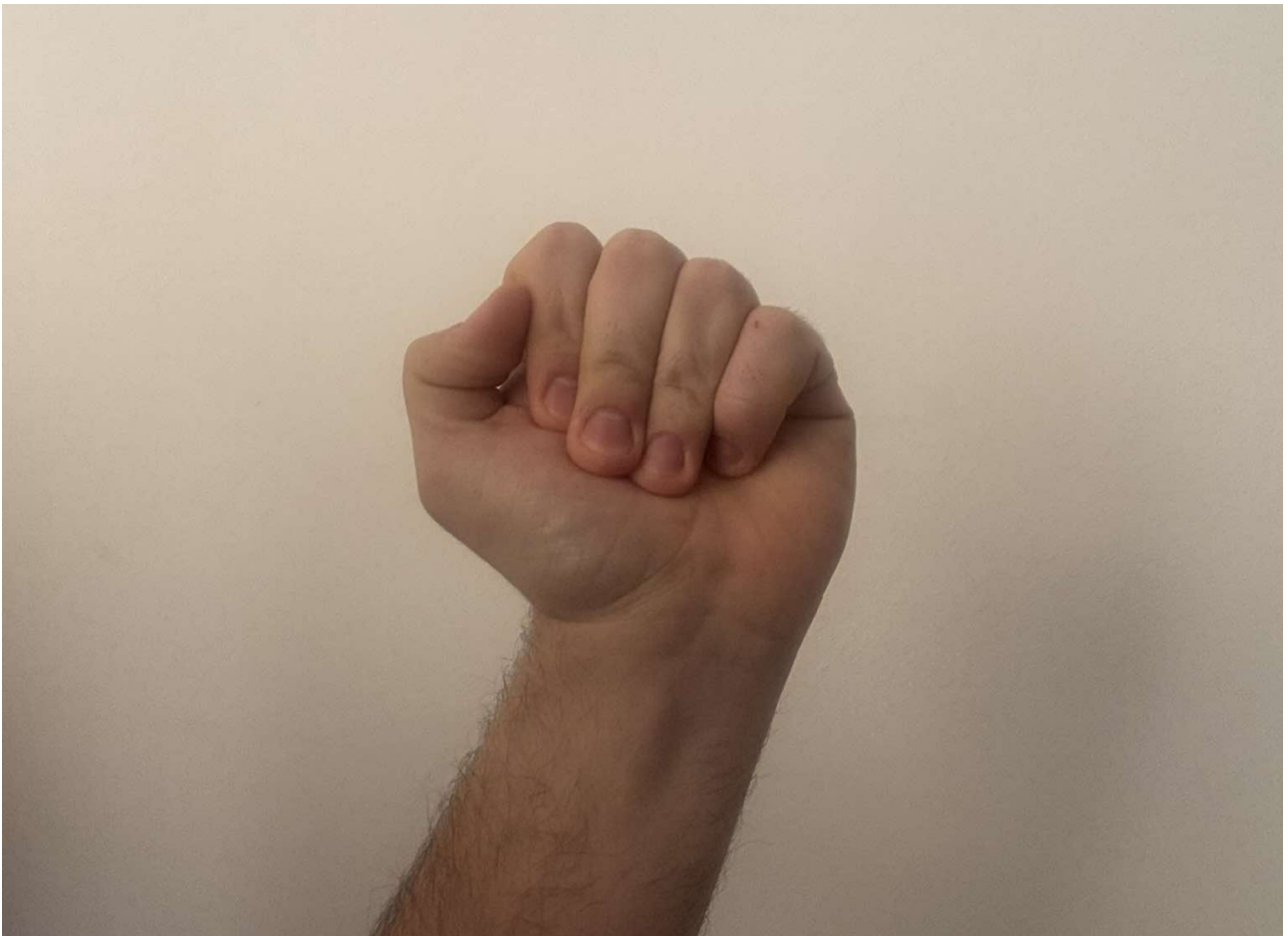
- Wskazanie w prawo - następny slajd,



- Wskazanie w lewo - poprzedni slajd,



- Zaciśnięta pięść - zmiana narzędzi.



### 3.3 Narzędzia do rysowania

Dostępne są następujące narzędzia:

- Ołówek - rysowanie po slajdach,
- Gumka - usuwanie rysunków,
- Wskaźnik laserowy - podkreślanie elementów prezentacji.

## 4 Implementacja

### 4.1 Detekcja gestów

```
1  const detectGesture = (landmarks) => {
2    if (!landmarks || landmarks.length === 0) return null;
3
4    const hand = landmarks[0];
5    const wrist = hand[0];
6    const thumbTip = hand[4];
7    const indexTip = hand[8];
8    const middleTip = hand[12];
9
10   // Wykrywanie gestu wskazywania
11   const isPointing =
12     indexTip.y < middleTip.y &&
13     Math.abs(thumbTip.y - wrist.y) < 0.1;
14
15   // Wykrywanie gestu pięści
16   const isFist = indexTip.y > wrist.y - 0.1;
17
18   if (isPointing) {
19     return {
20       type: 'pointing',
21       position: { x: indexTip.x, y: indexTip.y }
22     };
23   }
```

```

24 if (isFist) return 'fist';
25
26 return null;
27 };

```

## 4.2 Wymagania programowe

- Nowoczesna przeglądarka internetowa (Chrome 88+, Firefox 85+, Safari 14+)
- Obsługa WebGL dla MediaPipe
- Włączona obsługa JavaScript
- Dostęp do kamery internetowej
- Minimalna rozdzielczość kamery: 640x480
- Stabilne połączenie internetowe (dla ładowania modeli MediaPipe)

## 5 Instalacja i uruchomienie

### 5.1 Za pomocą Docker

1. Pobierz najnowszą wersję obrazu Dockera z repozytorium na GitHubie: <https://github.com/ZegarekPL/Gesture-Controlled-Presentation-Tool---Frontend/releases>

2. Załaduj obraz do Dockera:

```
1 docker load -i gesture_controlled_presentation_tool_frontend-v0.1.1.tar
```

3. Uruchom kontener:

```
1 docker run -d -p 3000:3000 gesture_controlled_presentation_tool_frontend:v0.1.1
```

4. Aplikacja będzie dostępna pod adresem: <http://localhost:3000>

### 5.2 Instalacja lokalna

#### 5.2.1 Wymagania

- Node.js 18+
- npm lub yarn
- Git

#### 5.2.2 Kroki instalacji

1. Sklonuj repozytorium:

```

1 git clone https://github.com/ZegarekPL/Gesture-Controlled-Presentation-Tool---Frontend.git
2 cd Gesture-Controlled-Presentation-Tool---Frontend

```

2. Zainstaluj zależności:

```

1 npm install
2 # lub
3 yarn install

```

3. Uruchom aplikację w trybie deweloperskim:

```

1 npm run dev
2 # lub
3 yarn dev

```