

Ratatosk Projekt Logbuch - Thematisch sortiert und überarbeitet

Projekt: Ratatosk - Kommunikationshilfe für Menschen mit Behinderungen

Entwicklungszeitraum: Dezember 2024 - Januar 2025

Technologie: Vue.js 3, TypeScript, Tailwind CSS, MediaPipe

Inhaltsverzeichnis

- [1. Projektarchitektur und Setup](#)
- [2. Gesichtserkennung und Kamera-Integration](#)
- [3. Benutzeroberfläche und Design](#)
- [4. Kommunikationssystem](#)
- [5. Accessibility und Barrierefreiheit](#)
- [6. Einstellungen und Konfiguration](#)
- [7. Fehlerbehebung und Optimierung](#)

Projektarchitektur und Setup {#projektarchitektur}

Moderne Vue.js 3 Architektur

Das Ratatosk-Projekt wurde von Grund auf mit modernen Web-Technologien neu entwickelt. Die ursprüngliche Vue 2 Struktur wurde durch eine vollständig typisierte Vue 3 Anwendung ersetzt.

Kernkomponenten:

- Vue 3 mit Composition API:** Moderne Reaktivität und bessere Performance
- TypeScript:** Vollständige Typisierung für bessere Code-Qualität
- Vite Build System:** Schnelle Entwicklung und optimierte Produktions-Builds
- Feature-basierte Architektur:** Organisierte Ordnerstruktur nach Funktionsbereichen

Projektstruktur:

```
src/
├── features/           # Feature-Module
│   ├── communication/ # Kommunikationsfunktionen
│   ├── face-recognition/ # Gesichtserkennung
│   ├── settings/       # Einstellungen
│   └── navigation/     # Navigation
├── shared/            # Gemeinsame Komponenten
├── core/              # Kern-Funktionalitäten
└── types/             # TypeScript Definitionen
```

State Management:

- Pinia Stores:** Moderne State-Verwaltung für Einstellungen, Kommunikation und Gesichtserkennung
- Persistierung:** Automatische Speicherung in localStorage
- Reaktivität:** Vollständig reaktive Datenbindung

Build und Deployment:

- Vite:** Moderne Build-Pipeline mit Hot Module Replacement

- **TypeScript:** Vollständige Typisierung mit strict mode
 - **ESLint:** Code-Qualität und Konsistenz
 - **GitHub Pages:** Automatisches Deployment
-

Gesichtserkennung und Kamera-Integration {#gesichtserkennung}

MediaPipe Face Recognition System

Das Herzstück der Anwendung ist die Integration von Google's MediaPipe für präzise Gesichtserkennung und Blinzeln-Erkennung.

Technische Implementierung:

- **MediaPipe Face Mesh:** 468 Gesichtspunkte für präzise Erkennung
- **Eye Detection:** Separate Erkennung für linkes und rechtes Auge
- **Blink Detection:** Konfigurierbare Schwellenwerte für Blinzeln-Erkennung
- **Performance:** Optimiert für 10 FPS auf verschiedenen Geräten

Browser-Kompatibilität:

- **Safari-Support:** Spezielle Behandlung für Safari-Browser
- **Fallback-Strategien:** Mehrere Kamera-Konfigurationen
- **Error Handling:** Robuste Fehlerbehandlung bei Kamera-Problemen

Konfigurierbare Parameter:

- **Blink Sensitivity:** 0.7 Sekunden (Standard)
- **Detection Confidence:** 0.5 (anpassbar)
- **Tracking Confidence:** 0.5 (anpassbar)
- **Face Factor:** 55 (Schwellenwert für Augenverschluss)

Landmarks-System:

```
const EYE_LANDMARKS = {  
  LEFT_EYE: { TOP: 386, BOTTOM: 374, LEFT: 263, RIGHT: 362 },  
  RIGHT_EYE: { TOP: 159, BOTTOM: 145, LEFT: 33, RIGHT: 133 }  
}
```

Kamera-Integration und Berechtigungen

getUserMedia API Integration:

- Automatische Berechtigungsanfrage für Kamera-Zugriff
- Fallback-Strategien bei fehlenden Berechtigungen
- Robuste Error Handling für verschiedene Browser

Safari-spezifische Optimierungen:

- Spezielle Video-Attribute für Safari-Kompatibilität
 - `playsinline` und `webkit-playsinline` Support
 - Muted Video für automatische Wiedergabe
-

Benutzeroberfläche und Design {#benutzeroberflaeche}

Responsive Grid-System

Das Anwendungsdesign basiert auf einem konfigurierbaren 3x2 Grid-System mit zentriertem Layout.

Grid-Konfiguration:

- **Kachel-Größe:** 422px × 422px
- **Icon-Größe:** 125px × 125px
- **Text-Größe:** 40px
- **Abstände:** 32px zwischen Kacheln
- **Border-Radius:** 10px für moderne Optik

Zentriertes Layout:

- **Flexbox:** `items-center justify-center` für perfekte Zentrierung
- **Responsive Design:** Funktioniert auf allen Bildschirmgrößen
- **SVG-Icons:** Hochauflösende Icons für alle Kacheln

Dark Mode Implementation

Toggle-Button:

- Position: Oben rechts im Header
- Icons: Sonne für Light Mode, Mond für Dark Mode
- Hover-Effekte: `hover:bg-gray-300 dark: hover:bg-gray-600`

Dark Mode Styles:

- **Hintergrund:** `dark:bg-gray-900` für Hauptcontainer
- **Header:** `dark:bg-gray-800` für Header
- **Text:** `dark:text-white` für alle Texte
- **Kacheln:** `rgba(55,65,81,0.3)` Hintergrund im Dark Mode
- **Borders:** Weiße Borders im Dark Mode
- **Icons:** `dark:invert-0` für korrekte Icon-Farben

Startseite und Navigation

Startseite-Features:

- **Kamera-Status:** Visueller Status der Kamera mit Farbcodierung
- **Benutzerführung:** Klare Anweisungen für den User
- **Error Handling:** Robuste Fehlerbehandlung bei Kamera-Problemen
- **Responsive Design:** Funktioniert auf allen Geräten

Navigation-System:

- **Router:** Vue Router 4 mit TypeScript
- **Route Guards:** Automatische Kamera-Aktivierung
- **Breadcrumbs:** Klare Navigation zwischen Seiten

Kommunikationssystem {#kommunikationssystem}

Virtuelle Tastatur mit QWERTZ-Layout

Tastatur-Features:

- **QWERTZ-Layout:** Deutsche Tastaturbelegung
- **Blinzeln-Navigation:** Automatische Navigation durch Blinzeln
- **Visuelle Hervorhebung:** Aktive Tasten werden hervorgehoben
- **Text-Speicherung:** Persistente Text-Speicherung
- **TTS-Integration:** Automatische Sprachausgabe

Navigation-System:

- **Automatische Navigation:** Blinzeln-basierte Tastatur-Navigation
- **Zurück-Button:** Integrierter Zurück-Button in der Tastatur
- **Silben-Tasten:** Spezielle Tasten für häufige Silben
- **Löschen-Funktion:** Rechte Maustaste als Blinzeln-Ersatz

Text-to-Speech (TTS):

- **Vollständige TTS-Integration:** Sprachausgabe für alle Eingaben
- **Menü-Navigation:** Sprachausgabe für alle Menüpunkte
- **Konfigurierbarkeit:** Ein-/Ausschaltbare TTS-Funktion
- **Debug-Informationen:** TTS-Debug für Entwicklung

Kommunikationsseiten

Ich-Seite:

- **5 Kacheln:** Strukturierte Kommunikationsmöglichkeiten
- **Design-Konsistenz:** Angepasst an Hauptseite-Design
- **Icon-Integration:** Korrekte Icon-Farben und Filter
- **Responsive Layout:** Funktioniert auf allen Geräten

Gefühle-Seite:

- **Emotionen-Auswahl:** Strukturierte Gefühls-Auswahl
- **Vergrößerte Ansicht:** Bessere Sichtbarkeit der Optionen
- **Auswahl-Anzeige:** Visuelle Bestätigung der Auswahl
- **TTS-Integration:** Sprachausgabe für alle Gefühle

Accessibility und Barrierefreiheit {#accessibility}

Behindertengerechte Bedienung

Blinzeln-Steuerung:

- **Robuste Erkennung:** Verbesserte Blinzeln-Erkennung
- **Konfigurierbare Sensibilität:** Anpassbare Erkennungsschwellen
- **Konsistente Navigation:** Einheitliche Blinzeln-Erkennung auf allen Seiten
- **Performance-Optimierung:** Optimierte Erkennungsalgorithmen

Accessibility-Features:

- **TTS-Integration:** Vollständige Sprachausgabe
- **High Contrast:** Unterstützung für hohen Kontrast
- **Large Text:** Unterstützung für große Schrift
- **Reduced Motion:** Unterstützung für reduzierte Animationen

Alternative Eingabemethoden:

- **Rechte Maustaste:** Als Blinzeln-Ersatz implementiert
- **Keyboard Navigation:** Vollständige Tastatur-Navigation
- **Touch Support:** Touch-Geräte-Unterstützung

Warnsystem und Alarm

Warning-Seite:

- **Piepton-Alarm:** Akustischer Alarm bei Problemen
- **Kontinuierlicher Alarm:** Anhaltender Alarm bei kritischen Situationen
- **Schriftart-Optimierung:** Bessere Lesbarkeit
- **Navigation:** Automatische Zurück-Navigation nach 10 Sekunden

Alarm-Features:

- **ServiceGlocke.wav:** Professioneller Alarm-Sound
 - **Visuelle Warnung:** Klare visuelle Warnungen
 - **Automatische Deaktivierung:** Intelligente Alarm-Deaktivierung
-

Einstellungen und Konfiguration {#einstellungen}

Settings Store System

Zentrale Einstellungsverwaltung:

- **UserSettings Interface:** Vollständige Typisierung der Einstellungen
- **Persistierung:** Automatische Speicherung in localStorage
- **Reaktivität:** Vollständig reaktive Einstellungen
- **Reset-Funktion:** Zurücksetzen auf Standardwerte

Standardkonfiguration:

- **Leuchtdauer:** 3000ms (3 Sekunden)
- **Blinzeln-Sensibilität:** 0.7 Sekunden
- **Auto-Modus:** 3000ms (3 Sekunden)
- **Sound:** Aktiviert
- **Voice:** Deaktiviert

Theme-Management:

- **Light/Dark Mode:** Automatischer Wechsel
- **Auto-Detection:** Automatische Erkennung der Systemeinstellungen
- **Persistierung:** Speicherung der Theme-Präferenzen

Konfigurierbare Parameter

Blinzeln-Einstellungen:

- **Sensibilität:** 0.7 Sekunden (Standard)
- **Dauer:** 2 Sekunden (Standard)
- **Geschwindigkeit:** 2 (Standard)
- **Auto-Modus:** 3000ms (Standard)

Accessibility-Einstellungen:

- **High Contrast:** Deaktiviert (Standard)
- **Large Text:** Deaktiviert (Standard)

- **Reduced Motion:** Deaktiviert (Standard)
-

Fehlerbehebung und Optimierung {#fehlerbehebung}

Code-Qualität und Syntax

TypeScript-Integration:

- **Vollständige Typisierung:** Alle Komponenten und Funktionen typisiert
- **Strict Mode:** Strengste TypeScript-Konfiguration
- **Interface-Definitionen:** Vollständige Interface-Definitionen
- **Error Handling:** Robuste Fehlerbehandlung

Syntax-Fehler-Behebung:

- **Kamera-Code:** Alle Syntax-Fehler in Kamera-Integration behoben
- **Vue-Templates:** Korrekte Template-Syntax
- **TypeScript-Errors:** Alle TypeScript-Fehler behoben
- **Build-Optimierung:** Optimierte Build-Konfiguration

Performance-Optimierung

Face Recognition Performance:

- **10 FPS:** Optimiert für 10 Frames pro Sekunde
- **Landmarks-Optimierung:** Effiziente Landmarks-Berechnung
- **Memory Management:** Optimierte Speicherverwaltung
- **Browser-Kompatibilität:** Optimiert für verschiedene Browser

Build-Optimierung:

- **Vite:** Moderne Build-Pipeline
- **Code-Splitting:** Automatisches Code-Splitting
- **Asset-Optimierung:** Optimierte Asset-Verwaltung
- **Bundle-Size:** Minimierte Bundle-Größe

Deployment und GitHub Pages

GitHub Pages Integration:

- **Automatisches Deployment:** Automatisches Deployment bei Git-Push
- **Dist-Ordner:** Build-Ordner für GitHub Pages
- **Asset-Optimierung:** Optimierte Assets für Produktion
- **Error Handling:** Robuste Fehlerbehandlung bei Deployment







Repository-Management:

- **Git-Historie:** Saubere Git-Historie mit aussagekräftigen Commits
 - **Branch-Strategy:** Main-Branch für Produktion
 - **Documentation:** Vollständige README und Logbuch
 - **Versioning:** Semantische Versionierung
-

Zusammenfassung und Projektstatus

Projekt-Status:  **Vollständig abgeschlossen**

Hauptfunktionen:

-  **Gesichtserkennung:** MediaPipe-basierte Erkennung mit 468 Landmarks
-  **Blinzeln-Steuerung:** Robuste Blinzeln-Erkennung mit konfigurierbaren Schwellenwerten
-  **Virtuelle Tastatur:** QWERTZ-Layout mit TTS-Integration
-  **Dark Mode:** Vollständige Dark Mode-Unterstützung mit Toggle-Button
-  **Accessibility:** Behindertengerechte Bedienung mit TTS und alternativen Eingabemethoden
-  **Responsive Design:** Funktioniert auf allen Geräten und Bildschirmgrößen

Technische Highlights:

- **Vue 3 + TypeScript:** Moderne Entwicklungsumgebung mit vollständiger Typisierung
- **MediaPipe:** Google's Gesichtserkennungs-API für präzise Erkennung
- **Tailwind CSS:** Utility-first CSS Framework für konsistentes Design
- **Pinia:** Moderne State-Verwaltung mit Persistierung
- **Vite:** Moderne Build-Pipeline mit Hot Module Replacement

Deployment:

- **GitHub Pages:** Live unter <https://leogisorb.github.io/ratatosk.2.0/>
- **Git Repository:** <https://github.com/leogisorb/ratatosk.2.0.git>
- **Dokumentation:** Vollständige README und thematisch sortiertes Logbuch

Zielgruppe: Das Ratatosk-Projekt richtet sich an Menschen mit Behinderungen, die auf alternative Kommunikationsmethoden angewiesen sind. Die Anwendung ermöglicht es, durch Blinzeln zu navigieren und zu kommunizieren, was eine wichtige Unterstützung für Menschen mit eingeschränkter Mobilität darstellt.

Erstellt am: 2025-01-31

Entwickler: Leopold Brosig

Projekt: Ratatosk - Kommunikationshilfe für Menschen mit Behinderungen