

THEREMINATOR

Technische Dokumentation

Emir Kokumov	2343993
Selim Ariguib	2288531
Said-Reza Waezsada	2164201

URL des git-Repositories zum Projekt:

<https://github.com/HarryGuib/Thereminator>

Inhaltsverzeichnis

<u>Ziel dieser Dokumentation</u>	3
<u>Benutzeroberfläche</u>	3
<u>Bedienungsanleitung</u>	4
<u>Programmiertechnische Umsetzung</u>	5
<u>Fazit</u>	6

Ziel dieser Dokumentation

Diese Dokumentation dient als komplementäres Dokument zu der bereits abgegebenen Konzept Dokumentation.

Im Rahmen dieser Dokumentation werden Ihnen Punkte erläutert, wodurch die Nutzung der Anwendung vereinfacht werden soll. Darüber hinaus wird erklärt, wie die Software umgesetzt wurde.

Hierbei wird zunächst die Benutzeroberfläche betrachtet und die Optionen, die dem Nutzer zur Verfügung stehen, präsentiert.

Anschließend wird eine Anleitung für die Anwendung zur Verfügung gestellt, um eine einfachere Nutzung sicher zu stellen.

Danach wird erläutert, wie die Anwendung programmiertechnisch umgesetzt wurde und welche Schnittstellen miteinander interagieren.

Benutzeroberfläche

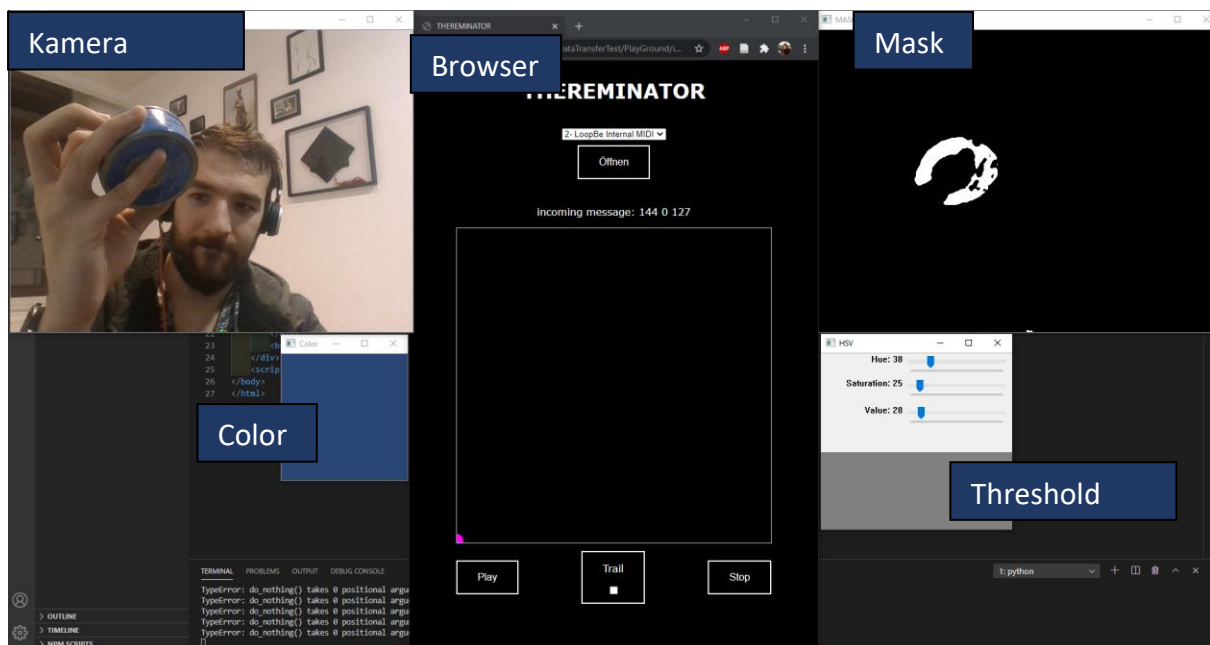


Bild 1: Benutzer Oberfläche

In diesem Bild sehen wir, was dem Nutzer beim Start der Anwendung entgegenkommt.

Zur Vereinfachung wurden die einzelnen Fenster mithilfe von blauen Rahmen benannt und deren Funktion erklärt.

Kamera

Hier sehen wir die Live-Aufnahme der Kamera. Hier hat der Nutzer die Möglichkeit die Erkennungsfarbe zu bestimmen. Dafür muss der Benutzer im Kamera Fenster auf die gewünschte Farbe klicken (z.B. man klickt auf das Bild einer Zitrone)

Color

In diesem Fenster wird die aktuell ausgewählte Suchfarbe angezeigt.

Threshold

Hier werden Toleranzen für die Farberkennung gewählt.

Mask

Hier wird die vom Programm erkannte Farben parallel zum Kamerabild angezeigt. Die erkannten Farben werden als weiße Silhouette vor einem schwarzen Hintergrund dargestellt.

Browser

Hier wird die Koordinate des Markers dargestellt und der Ton des Theremins ausgegeben.

- Der Play-Button startet die Tonausgabe, wobei deren Frequenz und Lautstärke abhängig von der Koordinate des Markers ist.
- Der Stop-Button stoppt die Tonausgabe der Anwendung.
- Die Trail Checkbox bietet eine Visuelle dekorative Funktion, indem die Änderung der Marker Koordinaten als Verlauf angezeigt wird.

Das Layout ist automatisch nach Bildschirmgröße skalierbar.

Bedienungsanleitung

- Nach dem Start der Anwendung stellt der Nutzer ein beliebiges Marker-Objekt vor der Kamera und klickt im Kamera-Fenster auf das Bild des Objekts.
- Hierbei ist es gleichgültig, was es für ein Objekt ist.
- Anschließend kann man im Mask-Fenster überprüfen, ob der Marker wie gewünscht erkannt wird oder nicht.
- Man hat die Möglichkeit mithilfe des Threshold-Fensters die Toleranzen für die Farberkennung einzurichten.
- Wenn nun die Maske dem Wünschen des Nutzers entspricht, kann dieser nun den Play-Button des Browser-Fensters anklicken, um die Tonausgabe zu starten.

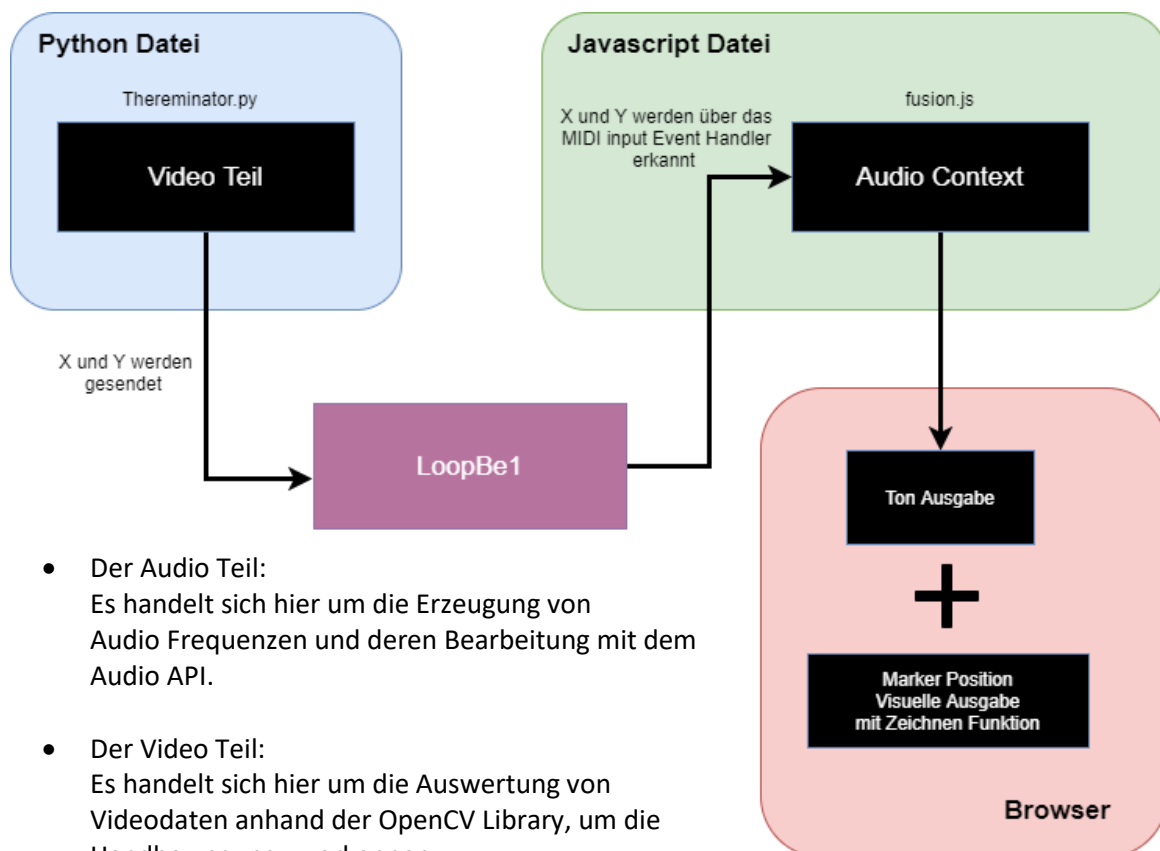
(Achtung: Sicher stellen, dass der LoopBe1 Driver nicht gemutet ist)

- Hier wird die Frequenz des abgespielten Tons von der x-Koordinate und die Lautstärke von der y-Koordinate bestimmt.
- Der Nutzer kann den Marker vor seine Kamera bewegen, um verschiedene Töne abzuspielen.
- (Optional) Um ein visuelles Spektakel zu erleben, kann der Nutzer den Trail-Checkbox aktivieren, der den Verlauf des Markers darstellt.
- Zur Beendigung der Anwendung kann der Nutzer auf den Stop Button klicken.

Programmiertechnische Umsetzung

Technische Umsetzung

Das Projekt ist grundlegend in 3 Baustellen aufgeteilt:



- **Der Audio Teil:**
Es handelt sich hier um die Erzeugung von Audio Frequenzen und deren Bearbeitung mit dem Audio API.
- **Der Video Teil:**
Es handelt sich hier um die Auswertung von Videodaten anhand der OpenCV Library, um die Handbewegung zu erkennen.
- **Die Verbindung zwischen Audio und Video:**
Schließlich werden die beiden oben genannten Bausteine zusammenverbunden, um anhand der Bewegung (bzw. Koordinaten der Handposition, die von der Video-Datei ausgewertet werden), die die entsprechenden Frequenzen von der Audio-Datei erzeugt.
Das Ganze geschieht durch den LoopBe1 Driver, was MIDI-Daten von Python an JavaScript schickt. In unserem Projekt senden wir X und Y Koordinaten des Markes von dem Python Datei: „Thereminator.py“ an die JavaScript Datei „Fusion.js“ und das anhand eines MIDI Kanals.

Aufwand der Teamarbeit

In der Konzeptbeschreibung hatte das Projekt einen Umfang von 180 Personenstunden. In den ersten Wochen haben wir laut dem Plan alle geteilt an eigenen Aufgaben gearbeitet, was uns immer zur Verzögerung von unseren Fristen geführt hatte. Also zum 5. Januar hatten wir keine richtigen Ergebnisse. Um zum 19. Januar unser Projekt fertig zu kriegen, haben wir uns entschieden, uns die restlichen 2 Wochen 4 mal die Woche je 3 Stunden per Team-Calls an dem Projekt zu arbeiten. Die Entscheidung hat unsere Arbeit effizienter gemacht, und jede Session haben wir mehrere explizite Fortschritte gemacht. Das neue Vorgehen zur Arbeit hat uns motiviert, neue Features, wie z. B. Zeichnen mithilfe von bewegten Objekten, zu implementieren. Somit haben wir 7 Wochen statt erwarteten 6 Wochen und 222 Personenstunden statt 180 gebraucht.

Planung

Gruppenmitglied	Aufgaben bis zum 05.01	Aufgaben nach dem 05.01
Emir	Hauptsächlich Video Teil und HTML: <ul style="list-style-type: none">- Tracking Funktion der hand- Optimieren	<ul style="list-style-type: none">- Audio- und Video Teil zusammen durcharbeiten- Wechselung von Datenübertragungsbibliothek von "eel" auf "MIDI"- Neue Features hinzufügen- Minimalistisches Design für die Web-App anpassen
Reza	Das Mergen von den beiden Teilen <ul style="list-style-type: none">- Audio und Video zusammenbringen- Fertigstellung der Anwendung	
Selim	Hauptsächlich Audioteil und CSS : <ul style="list-style-type: none">- Anbindung der WebAudio API- Gestalten eines sinnvollen Layouts	

Fazit

Umsetzung von der Farbmarkierung benötigt gleichmäßige verteilte Beleuchtung, um präzise zu sein.

Durch MIDI lässt sich einfach verschiedene Datenformen übertragen.

Körperliche Interaktion erhöht den Spaßfaktor.

Die Arbeit im Team hat sehr viel Spaß gemacht!