

PRODUKT

# SoundGrid – Python

Marko Dufala, Adam Štecho  
Date: 5.1.2024



## SoundGrid App

|            |                  |
|------------|------------------|
| ☀ Status   | Done             |
| ➤ Course   | 👤 MM1            |
| 📅 Due date | @January 6, 2025 |

### Závěrečná správa – Obsah PDF

## 1. Úvod a zámer projektu

Moderní DJ-i využívají širokou škálu nástrojů na tvorbu hudby, od mixážních pultů po komplexní softvér, který umožňuje manipulaci so zvukom v reálnom čase. Tento projekt bol inšpirovaný kreatívnym procesom DJ-ov a ich schopnosťou vytvárať audiovizuálne zážitky. Naším cieľom bolo spojiť techniky spracovania obrazu a zvukovej syntézy a vytvoriť interaktívny nástroj, ktorý využíva kameru na ovládanie zvukových parametrov.

Projekt vznikol ako súčasť našej semestrálnej práce, pričom sme chceli ukázať praktické využitie programovania na spracovanie obrazu a zvuku. Výsledkom je aplikácia, ktorá umožňuje používateľom experimentovať s detekciou farieb a vytvárať zvuky na základe vizuálnych podnetov.

## 2. Teoretický návrh

Aplikácia je založená na troch hlavných moduloch:

### 1. Spracovanie obrazu (Image Processing):

- Kamera snímá obraz v reálnom čase.
- Detegujeme objekty na základe prednastaveného odtieňa (hue).
- Veľkosť detegovaného objektu sa používa na výpočet parametrov zvuku.

### 2. Zvuková syntéza (Sound Synthesis):

- Generátor zvuku je modulovaný parametrami získanými z obrazu.

- Podporujeme rôzne typy zvukov, ako sú sínusové oscilátory, FM syntéza alebo prehrávanie samplov.
- Efekty, ako napríklad reverb alebo filter, dodávajú zvuku dynamiku.

### 3. Používateľské rozhranie (UI):

- Vizualizácia spracovaného obrazu: kamera, maska farby a okraje.
- Ovládacie prvky na nastavenie zvukových parametrov (hlasitosti, typ syntézy, efekt reverb).

#### Schéma systému:

```
[ Kamera ] -> [ Detekcia odtieňa ] -> [ Parametre syntézy ] -> [ Generátor zvuku ] -> [ Výstup ]
```

## 3. Popis riešenia

### 3.1 Spracovanie obrazu

- Pomocou knižnice **OpenCV**:
  - Kamera sníma obraz a konvertuje ho do farebného priestoru HSV.
  - Maskujeme oblasti zodpovedajúce požadovanému odtieňu (hue) a vyhodnocujeme ich veľkosť.
  - Obrisy detegovaných objektov sú použité na vizuálnu aj zvukovú spätnú väzbu.
- Hlavné kroky:
  - Detekcia objektov podľa odtieňa a citlivosti.
  - Výpočet frekvencie na základe veľkosti detegovaného objektu.
  - Aktualizácia vizualizácií: kamera, okraje, maska.

### 3.2 Generovanie zvuku

- **Oscilátory a nástroje**:
  - Vytvorili sme niekoľko zvukových nástrojov (sínus, píla, FM syntéza, samplovanie).
  - Hlasitosť a frekvencia sú modulované na základe parametrov detekovaných z obrazu.
- **Efekty**:
  - Reverb a filtre pridávajú dynamiku a textúru zvukom.
- **Optimalizácia proti praskaniu zvuku**:
  - Fadery s jemným fade-in/fade-out pre hladké prechody.
  - Objekty sa nezastavujú okamžite, aby sa dokončilo hladké zakončenie.

Výsledný zvuk, ktorý je generovaný z tlačítok sme syntetizovali do jedného zvuku, aby sa odstránila distorzia zvuku, ktorá vznikala keď hralo viacero nástrojov naraz (najmä rovnaké nástroje)

### 3.3 Používateľské rozhranie

- **wxPython**:
  - Živý obraz z kamery s vizualizáciou detekovaných oblastí.

- Posuvníky na nastavenie odtieňa, citlivosti a zvukových parametrov.
  - Reálnečasová spätná väzba pre užívateľské ladenie.
- 

## 4. Návod na spustenie a ovládanie

### Požiadavky:

- **Python 3.9+**
- Nainštalované knižnice: `numpy`, `opencv-python`, `wxPython`, `pyo`.

### Spustenie aplikácie:

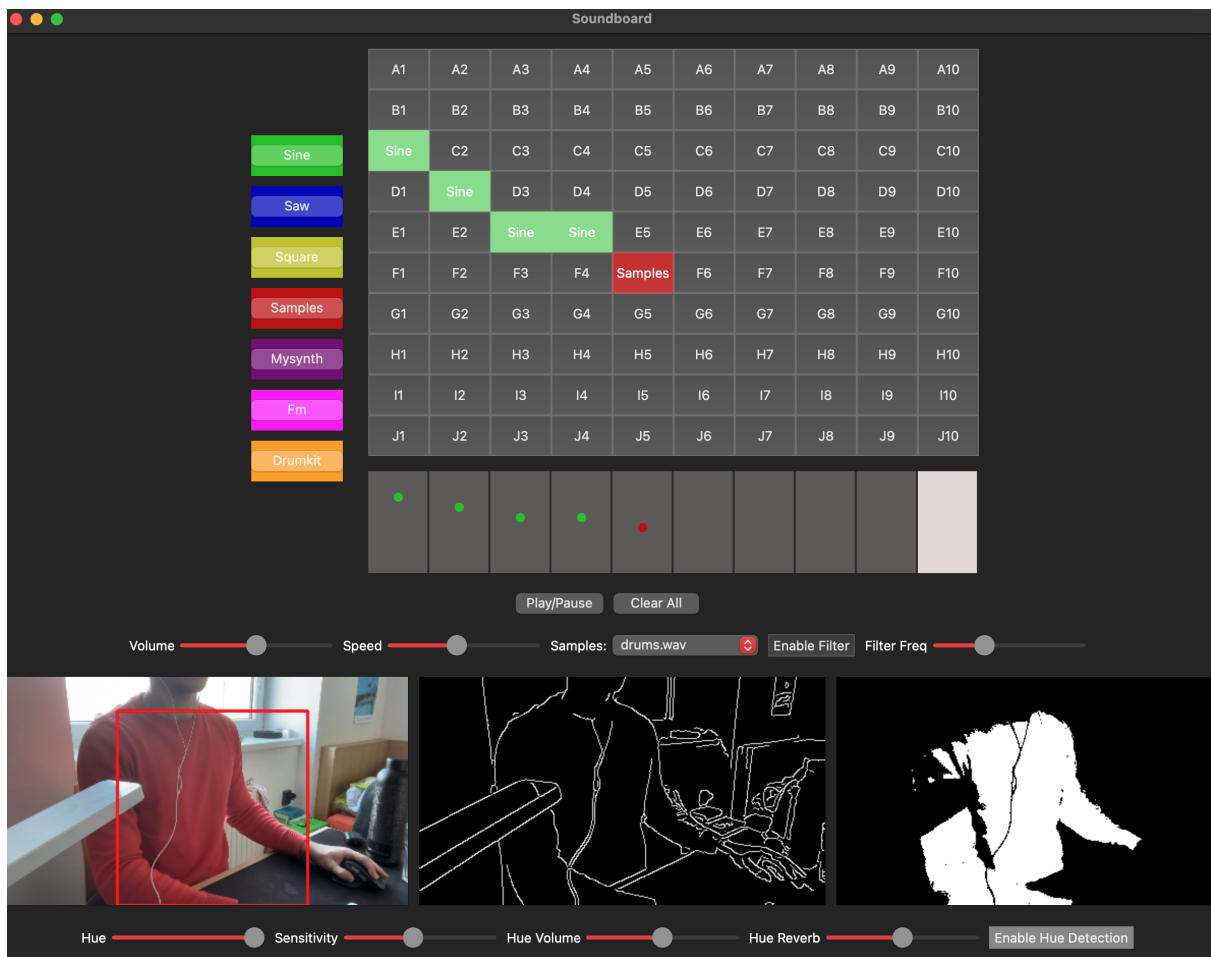
- Spustíte hlavný súbor:

```
python main.py
```

### Ovládanie aplikácie:

- **Výber odtieňa:**
    - Nastavte farbu, ktorú chcete detegovať, pomocou posuvníka.
  - **Citlivosť detekcie:**
    - Upravte šírku rozsahu odtieňov.
  - **Typ nástroja:**
    - Vyberte si zvukový nástroj (sínus, FM syntéza, samplovanie).
  - **Efekty:**
    - Pridajte reverb alebo upravte hlasitosť.
  - **Vizualizácia:**
    - Kamerový výstup so zvýraznenými oblasťami.
    - Zobrazenie okrajov a masky.
- 

### Screenshot aplikácie:



## 5. Výsledky a diskusia

- **Výstup zvuku:**
  - Generovaný zvuk reflektuje parametre detegovaných vizuálnych objektov.
  - Praskanie zvuku bolo eliminované jemnými prechodmi (fade-in/out).
- **Flexibilita aplikácie:**
  - Rozšíriteľné o ďalšie nástroje a efekty.
- **Vizualizácia:**
  - Jasná spätná väzba používateľovi, ktorá umožňuje doladenie nastavení.

## 6. Záver

Tento projekt demonštruje, ako je možné kreatívne využiť programovanie na prepojenie vizuálnych a zvukových prvkov. Výsledný nástroj má potenciál byť experimentálnym prostriedkom pre umelcov alebo študentov, ktorí chcú preskúmať základné princípy spracovania obrazu a syntézy zvuku.