

# Sprawozdanie

## "Utworzenie środowiska Python do montażu filmów"

11.06.2021

### Spis treści

<b>1</b>	<b>Cel projektu</b>	<b>2</b>
<b>2</b>	<b>Użyte biblioteki</b>	<b>2</b>
2.1	MoviePy . . . . .	2
2.2	numpy . . . . .	2
2.3	matplotlib . . . . .	3
2.4	librosa . . . . .	3
2.5	PyQt5 . . . . .	3
2.6	cv2 . . . . .	3
2.7	random . . . . .	3
<b>3</b>	<b>Konfiguracja stacji roboczej</b>	<b>3</b>
<b>4</b>	<b>Opis głównej biblioteki</b>	<b>4</b>
<b>5</b>	<b>Funkcje biblioteki MoviePy</b>	<b>4</b>
<b>6</b>	<b>Symboliczne drzewo programów</b>	<b>5</b>
<b>7</b>	<b>Wykonane programy</b>	<b>5</b>
7.1	Obrazy.ipynb . . . . .	5
7.2	Muzyka.ipynb . . . . .	5

7.3	Text.ipynb . . . . .	6
7.4	Czas w filmie, modyfikacja klatek.ipynb . . . . .	6
7.5	Wytwarzanie wideo z użyciem wykresów.ipynb . . . . .	6
7.6	Przykłady użycia maski (mask).ipynb . . . . .	6
7.7	Tracking - zamazywanie twarzy.ipynb . . . . .	6
7.8	Losowy automontaż.ipynb . . . . .	6
7.9	Przykładowy montaż filmu.ipynb . . . . .	6
7.10	Program z GUI do montowania filmu . . . . .	7
7.10.1	Krótką instrukcją działania: . . . . .	7
<b>8</b>	<b>Rezultaty</b>	<b>10</b>
<b>9</b>	<b>Wnioski</b>	<b>10</b>
9.1	Osiągnięte rezultaty . . . . .	10
9.2	Co można by usprawnić? . . . . .	10
<b>10</b>	<b>Słownik pojęć</b>	<b>11</b>
<b>11</b>	<b>Dokumentacja</b>	<b>11</b>

# 1 Cel projektu

Celem projektu było utworzenie środowiska Python do montażu filmów przy pomocy dostępnych bibliotek. Środowisko może służyć do generacji filmów w różnej formie i z różnymi efektami.

## 2 Użyte biblioteki

### 2.1 MoviePy

MoviePy to biblioteka pozwalająca na wykonywanie cięć, łączeń, przejść pomiędzy klipami wideo/zdjęciami. Umożliwia wstawianie napisów do filmów, czy muzyki.

### 2.2 numpy

Numpy to biblioteka służąca do używania różnych operacji matematycznych (np. sinus).

## 2.3 matplotlib

Matplotlib to biblioteka głównie służąca do rysowania wykresów matematycznych. Posłużyła do generacji animacji funkcji i wektorów.

## 2.4 librosa

Librosa to biblioteka służąca do zarządzania plikami muzycznymi i zawierająca narzędzia do ich obsługi.

## 2.5 PyQt5

Biblioteka służąca do generowania GUI (graficznego interfejsu użytkownika).

## 2.6 cv2

cv2 to biblioteka zawierająca kolekcję algorytmów służących do przetwarzania obrazów.

## 2.7 random

random posłużył to generacji liczb losowych

# 3 Konfiguracja stacji roboczej

Stacja robocza powinna być skonfigurowana przed rozpoczęciem pracy. Minimalne wymagania:

- Windows, Linux
- 4GB Ram (zalecane więcej)
- Python 3.8 + (używana była Anaconda 4.10.1)
- Powyższe biblioteki (można użyć instalatora `pip install #package-name`)

Opcjonalne dodatki:

- Jupyter Notebook (przydatne narzędzie wykorzystywane w data-science)

- Image-Magick (jeśli chcemy używać napisów w filmie)  
<https://www.imagemagick.org/script/index.php>

## 4 Opis głównej biblioteki

Z powyższych bibliotek najważniejsza jest biblioteka MoviePy, ponieważ dzięki niej następuje generacja filmów.

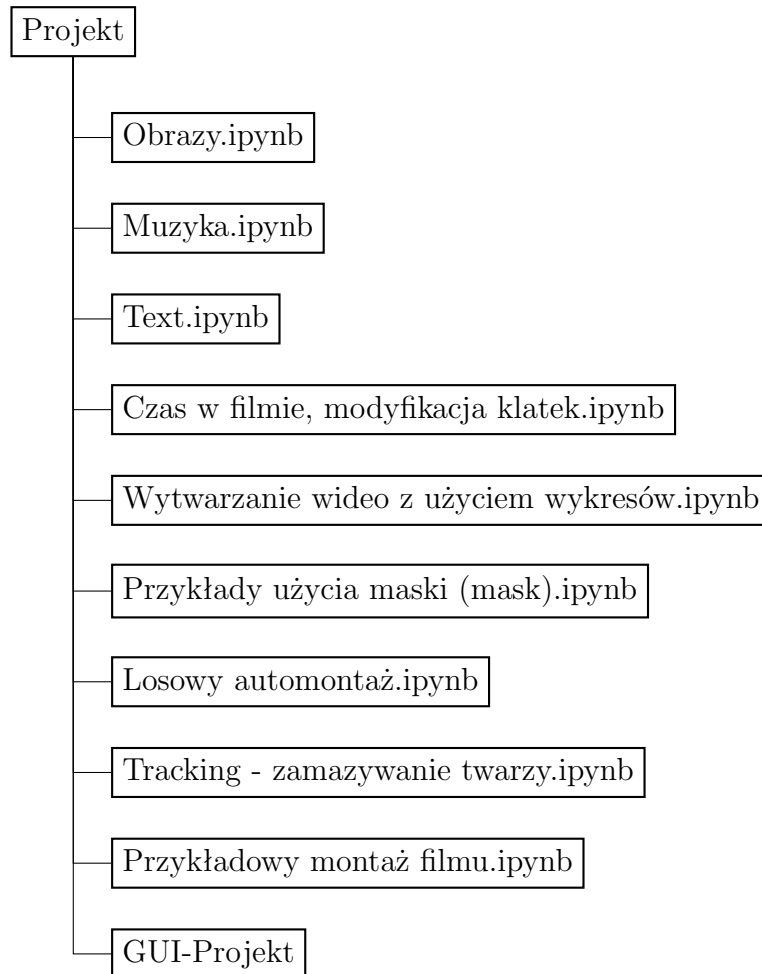
MoviePy to biblioteka pozwalająca na wykonanie zróżnicowanych funkcji multimedialnych poprzez programowanie. Umożliwia zarówno wykonywanie manipulacji obrazem typu: wykonywanie cięć, połączeń, wstawiania napisów, przetwarzania wideo (ang. processing), tworzenie własnych efektów specjalnych, jak i nieliniową edycję wideo, czy „zagnieżdżanie” kilku obrazów w jednym obrazie. Dodatkowo pozwala na zarządzanie i modyfikację dźwięków w zakresie wystarczającym dla wybranego tematu.

## 5 Funkcje biblioteki MoviePy

Przykładowe możliwości funkcji dostarczonych przez bibliotekę MoviePy:

1. Śledzenie i zamazywanie (ang. blur) twarzy danej osoby na wideo
2. Łączenie wielu klipów wideo/zdjęć w jeden film
3. Modyfikacja dźwięku w filmie (dodanie, usunięcie, zmiana)
4. Dodanie znaku wodnego/napisu do filmu
5. Stworzenie przejść pomiędzy klipami wideo/zdjęciami

## 6 Symboliczne drzewo programów



## 7 Wykonane programy

### 7.1 Obrazy.ipynb

Proste badania nad wykorzystaniem obrazów/klipów w montażu.

### 7.2 Muzyka.ipynb

Przykłady użycia muzyki, eksperymenty z muzyką.

### 7.3 Text.ipynb

Generowanie tekstu na filmie.

### 7.4 Czas w filmie, modyfikacja klatek.ipynb

Eksperymenty z zachowaniem się czasu w filmie. Wykonano zmianę czasu w filmie na podstawie wykresu. Przechwytywano także pojedyncze klatki i edytowano je (efekty graficzne, odbicie ramek)

### 7.5 Wytwarzanie wideo z użyciem wykresów.ipynb

Generowanie animacji funkcji matematycznych na filmie. Także napisałem krótki program, który robi teledysk - rysuje wektory w rytm do załączonej muzyki.

### 7.6 Przykłady użycia maski (mask).ipynb

Zastosowanie maskę wyciętą przy pomocy programu GIMP. Posłużyła ona do generacji tekstu, który przewija się pod nią, co daje ładny efekt wizualny.

### 7.7 Tracking - zamazywanie twarzy.ipynb

Implementacja trackingu. Należy podać współrzędne trackowania obiektu, a następnie na ekran nakładany jest obiekt śledzący (np. blur twarzy).

### 7.8 Losowy automontaż.ipynb

Implementacja pomysłu, gdzie użytkownik wybiera klipy wideo i muzykę, podaje parametry konfiguracyjne, a następnie program za pomocą losowości montuje cały film wraz z muzyką. Za każdym razem wychodzi inny film, co jest interesujące i można uzyskać ciekawy efekt końcowy.

### 7.9 Przykładowy montaż filmu.ipynb

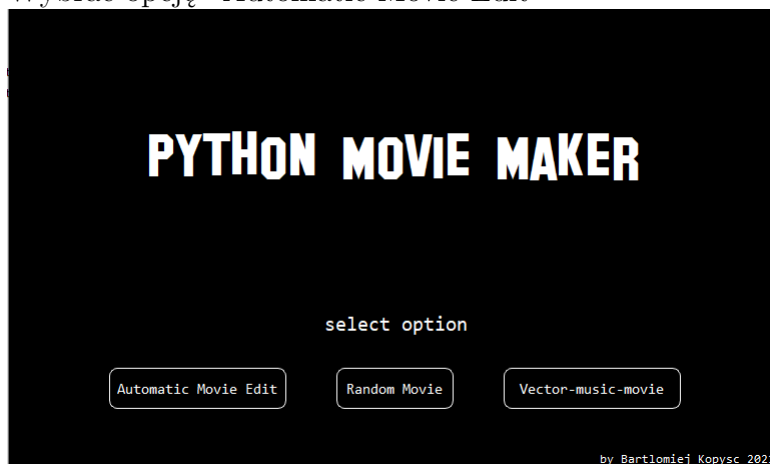
Ten program, to z góry obrany montaż filmu. Służy do pokazania, że można montować film bez graficznego programu do montażu, lecz używając tylko kodu i biblioteki *MoviePy*

## 7.10 Program z GUI do montowania filmu

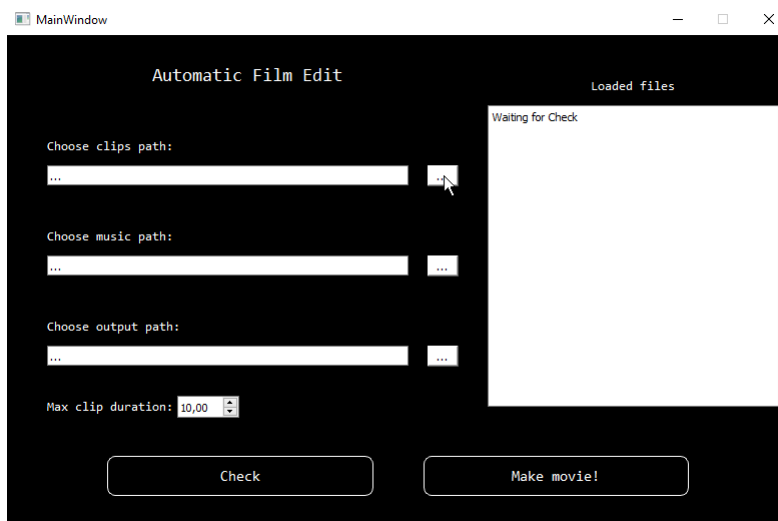
Koncept programu, gdzie zaimplementowane byłyby powyższe przykłady. Łatwy i czytelny dla użytkownika, mógłby mieć zastosowanie jako ciekawe narzędzie do osiągania różnych efektów z użyciem wideo i muzyki. Została zaimplementowana w nim metoda generacji filmu, na podstawie plików wybranych przez użytkownika (tzw. automontaż). Ingerencja użytkownika jest minimalna. GUI zostało zaprojektowane przy pomocy PyQt5.

### 7.10.1 Krótka instrukcja działania:

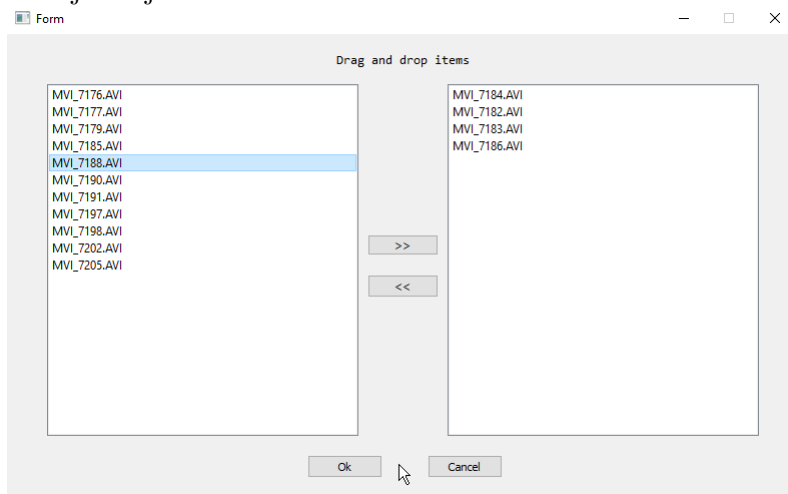
1. Należy uruchomić plik main.py (lub wykonany build)
2. Wybrać opcję "Automatic Movie Edit"



3. Uzupełnić pola ze ścieżkami do folderów

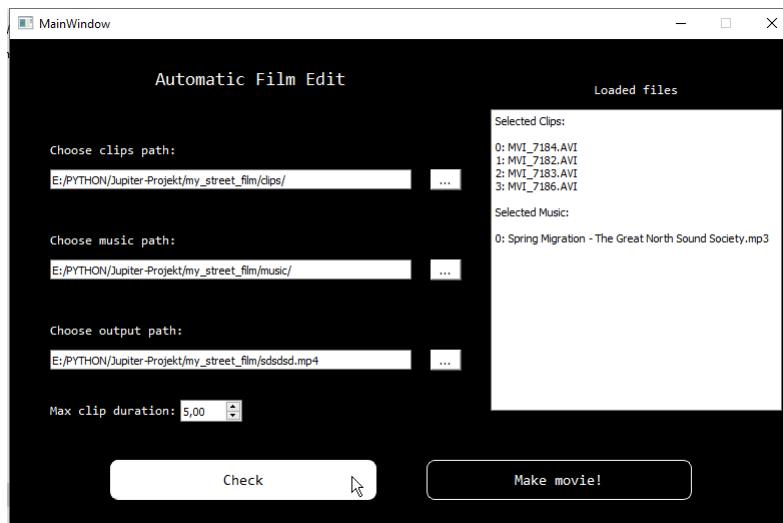


4. Poprzeciągać elementy (pliki), z lewego pola na prawę. Ułożyć w pożądaną kolejność

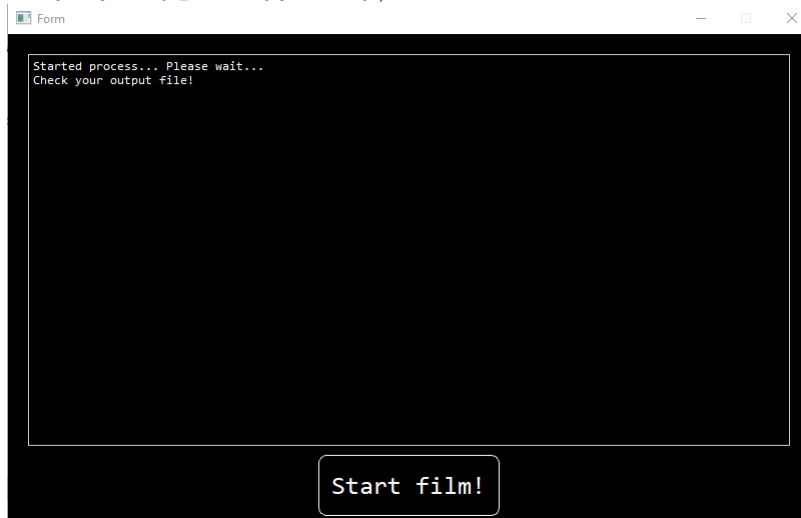


5. Wpisać maksymalny czas trwania jednego klipu
6. Kliknąć guzik "Check" w celu sprawdzenia załadowanych rzeczy





7. Wybrać przycisk "Make movie!"
8. Rozpocząć produkcję filmu przyciskiem "Start film!"
9. Odczekać na komunikat zwrotny, i przejść do wybranej lokalizacji (gdzie znajduje się plik wyjściowy)



10. Film jest już zmontowany.

## 8 Rezultaty

Rezultaty są widoczne w:

- Repozytorium GitHub (kod): <https://github.com/BKopysc/MoviePython>
- Serwisie YouTube (filmy): [link YouTube](#)

## 9 Wnioski

### 9.1 Osiągnięte rezultaty

Projekt przede wszystkim umożliwił:

- nauczenie się nowej biblioteki
- częściowe zrozumienie informacji na temat kodeków wideo
- naukę edycji obrazu, jako macierzy pikseli
- zapoznanie się z biblioteką PyQt5 i tworzenie GUI w Pythonie
- szersze zrozumienie języka Python

### 9.2 Co można by usprawnić?

Uważam, że można by usprawnić:

- Stworzenie pełnego GUI i implementacja różnych pomysłów
- Lepsze zarządzanie pamięcią RAM
- Zastosowanie wielowątkowości w celu przyspieszenia procesów
- Poprawa niektórych elementów kodu
- Większa znajomość języka i narzędzi do budowania GUI

## 10 Słownik pojęć

- Klip wideo - element będący częścią dłuższego nagrania.
- Znak wodny - częściowo przezroczysta (lub nie) warstwa/obiekt nałożony na obraz
- Przejście - efekt przy przełączeniu pomiędzy jednym klipem, a drugim. Najprostszym przejściem jest samo w sobie przełączenie. Bardziej zaawansowane przejścia są najczęściej stosowane w celu urozmaicenia filmu.
- Blur twarzy/obiektu - zamazanie obiektu na zdjęciu/wideo w celu ukrycia szczegółowych o nim informacji. Jest wiele algorytmów służące do tego celu, najczęściej używane jest rozmycie gaussowskie.
- Śledzenie obiektu - podążanie nałożonego elementu (np. blur) za określonym obiektem (zbiorem pikseli) na obrazie.
- Nieliniowa edycja wideo - dostęp do każdego fragmentu klipu w dowolnym czasie i z dowolnego miejsca podczas montażu.

## 11 Dokumentacja

- MoviePy: <https://zulko.github.io/moviepy>
- CV2: <https://pypi.org/project/opencv-python/>
- PyQt5: <https://pypi.org/project/PyQt5/>
- librosa: <https://librosa.org/doc/latest/index.html>
- matplotlib: <https://matplotlib.org>
- NumPy: <https://numpy.org>