

Trzaskowski Jakub – 163187 – sprawozdanie z projektu.

Link do Gita:

https://github.com/und3rthew4ter/Trzaskowski_projekt_na_zaliczenie_programowanie_.git

Cel projektu: Celem projektu było stworzenie aplikacji generującej muzykę w wybranym przez użytkownika nastroju. Są to krótkie, zapętlone melodyjki trwające 60 sekund, które po wybraniu przez użytkownika zapisywane są w tworzonym przez program folderze o nazwie „python_music”. Aplikacja ma prosty interfejs graficzny, a kod źródłowy został zoptymalizowany w porównaniu z początkowym stadium projektu.

Napotkane trudności: Głównym problemem było ułożenie nut w odpowiedniej kolejności w takim stylu, żeby odpowiadało to wybranemu nastrojowi.

Opis kodu i zasada jego działania:

Zastosowane biblioteki:

- **tkinter**: Biblioteka służąca do tworzenia elementów interfejsu graficznego (GUI).
- **messagebox**: Moduł umożliwiający wyświetlanie okienek z komunikatami dla użytkownika.
- **music21**: Biblioteka do pracy z notacją muzyczną, wykorzystywana tutaj do generowania plików MIDI.
- **os**: Moduł umożliwiający operacje na plikach i interakcję z systemem operacyjnym.

```
- #Trzaskowski Jakub - 163187  
- import tkinter as tk  
- from tkinter import messagebox  
- from music21 import stream, note, midi, chord  
- import os  
-
```

Funkcja `generate_music`:

- znajduje się w niej 5 pól wyboru z melodyjkami. Każda melodia ma 2 ścieżki dźwiękowe: nutki i bassy.

```
def generate_music(mood):  
    s = stream.Stream()  
  
    if mood == "Wesoły":  
        melody_pattern = ['C5', 'D5', 'E5', 'G5', 'A5', 'E5', 'G5', 'A5']  
        bass_pattern = ['C3', 'D3', 'E3', 'G3', 'A3']  
    elif mood == "Smutny":  
        melody_pattern = ['C4', 'D4', 'Eb4', 'G4', 'Ab4', 'G4', 'Ab4']  
        bass_pattern = ['C2', 'D2', 'Eb2', 'G2', 'Ab2']  
    elif mood == "Zły":  
        melody_pattern = ['E4', 'F4', 'G#4', 'B4', 'C5', 'F4', 'B4', 'C5', 'B4', 'C5']  
        bass_pattern = ['E2', 'F2', 'G#2', 'B2', 'C3']  
    elif mood == "Zmęczony":  
        melody_pattern = ['C4', 'E4', 'G4', 'B4', 'G4', 'B4', 'E4', 'E4']  
        bass_pattern = ['C2', 'E2', 'G2', 'B2']  
    elif mood == "Skupiony":  
        melody_pattern = ['C4', 'D4', 'F4', 'A4', 'D4', 'F4', 'D5', 'E5', 'G5']  
        bass_pattern = ['C2', 'D2', 'F2', 'A2', 'D5', 'E5', 'G8']
```

W poniższym fragmencie kodu zadeklarowany jest długość melodyjki (60s) i zapętlenie wybranego dźwięku do czasu zadeklarowanej długości, powielenie melodii i basów, generowanie strumienia muzycznego, określenie ścieżki zapisu tworzoną przez skrypt folderze. Komunikat o ścieżce po zapisaniu melodii oraz podanie ścieżki w outpucie kompilatora

```
min_duration = 60
note_duration = 0.25
num_repeats = int((min_duration / note_duration) / len(melody_pattern)) + 1

extended_melody_pattern = melody_pattern * num_repeats
extended_bass_pattern = bass_pattern * num_repeats

for p in extended_melody_pattern:
    s.append(note.Note(p, quarterLength=note_duration))

for p in extended_bass_pattern:
    s.append(note.Note(p, quarterLength=note_duration))

desktop_path = os.path.join(os.path.join(os.environ['USERPROFILE']), 'Desktop') # Windows
music_folder_path = os.path.join(desktop_path, 'Python_music')

if not os.path.exists(music_folder_path):
    os.makedirs(music_folder_path)

file_path = os.path.join(music_folder_path, f'{mood}_music.mid')

mf = midi.translate.music21ObjectToMidiFile(s)
mf.open(file_path, 'wb')
mf.write()
mf.close()

messagebox.showinfo("Informacja", f"Muzyka została wygenerowana i zapisana jako '{file_path}'.")
print(f"Plik zapisano w: {file_path}")
```

Funkcja on_button_click:

- Wywoływana jest po kliknięciu przycisku odpowiadającego wybranemu nastrojowi w GUIa następnie wywołuje opisaną powyżej funkcję „generate_music(mood)” w celu wygenerowania i zapisania muzyki na podstawie wybranego nastroju.

```
def on_button_click(mood):
    generate_music(mood)
```

W poniższym fragmencie kodu ustawiane są cechy okienka tkintera, czyli tytuł, wielkość i paleta barw. Następnie wyświetlany jest komunikat do użytkownika z instrukcją mówiącą, żeby wybrać melodię pasującą do jego aktualnego nastroju. Pod tym zostały utworzone przyciski dla każdej z emocji, a na końcu znajduje się uruchomienie pętli tkintera.

```
root = tk.Tk()
root.title("Generator muzyczki - Trzaskowski Jakub - nr.albumu:163187")
60 root.geometry("600x600")
root.configure(bg="orange")

label = tk.Label(root, text="Wybierz w jakim nastroju chcesz utworzyc muzyczkę:", bg="orange", fg="black", font=("Helvetica", 17))
label.pack(pady=20)

moods = ["Wesoly", "Smutny", "Zły", "Zmęczony", "Skupiony"]
70 for mood in moods:
    button = tk.Button(root, text=mood, command=lambda m=mood: on_button_click(m), bg="black", fg="orange", font=("Helvetica", 16), width=20)
    button.pack(pady=10)

root.mainloop()
```

Podsumowanie: Mój skrypt wykorzystuje Tkinter do stworzenia interfejsu użytkownika umożliwiającego wybór nastroju do generowania muzyki. Na podstawie wybranego nastroju generowane są krótkie zapętlone dźwięki w formacie MIDI. Wygenerowany plik jest zapisywany na pulpicie użytkownika w folderze "Python_music". Po zapisaniu, użytkownik otrzymuje informację o ścieżce zapisanego pliku za pomocą okienka dialogowego, a także jest to wyświetlane w konsoli.