1. Temat

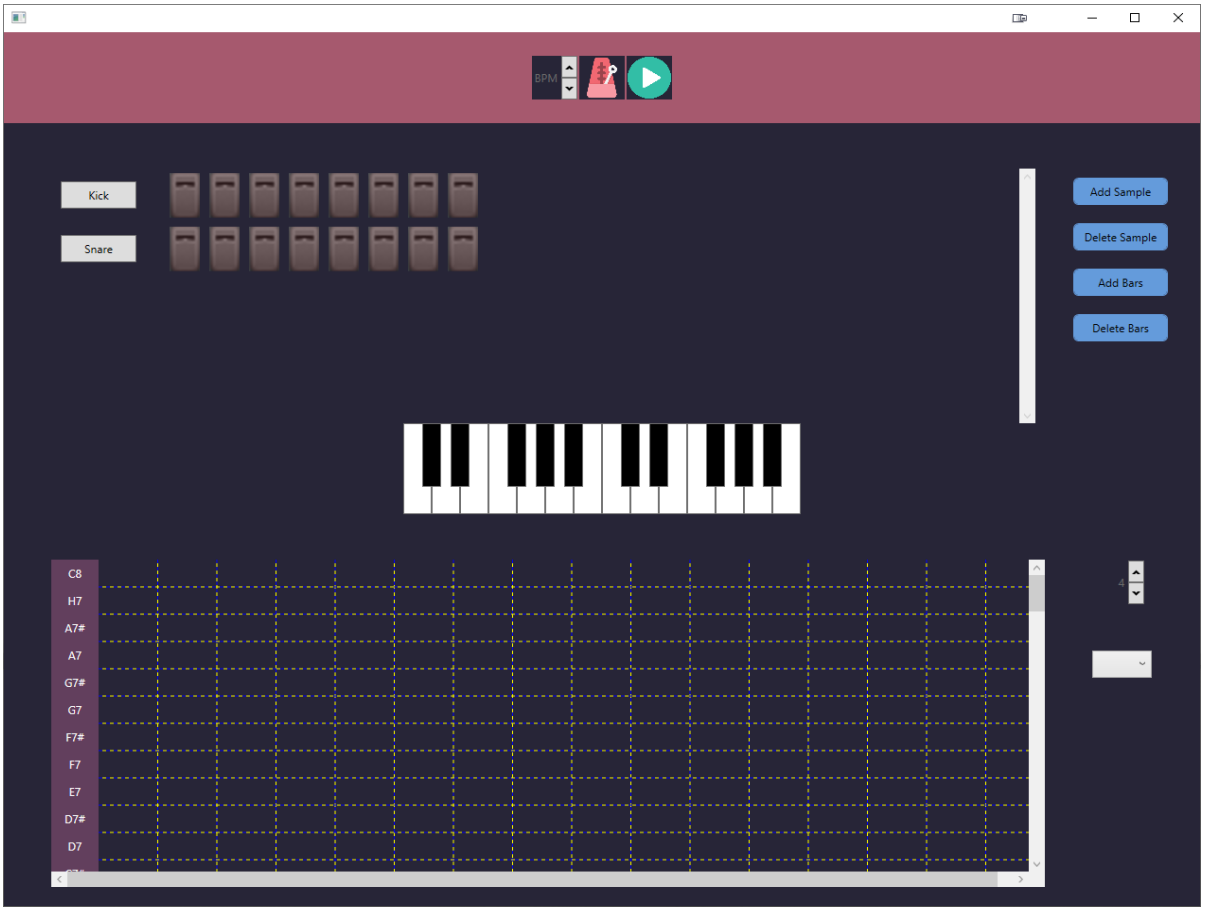
Tematem projektu jest stworzenie aplikacji okienkowej napisanej w języku C# przy pomocy frameworku WPF. Program ma być prymitywnym odpowiednikiem DAW (Digital Audio workspace) czyli oprogramowania służącego do szeroko rozumianej produkcji muzyki.

2. Analiza tematu

Program ma umożliwiać użytkownikowi odtwarzanie stworzonych przez siebie sekwencji plików dźwiękowych. Do jego dyspozycji jest sekwencer pozwalający na tworzenie prostych pętli perkusyjnych przy pomocy dowolnie wybranych plików Wav lub Mp3. Kolejna funkcjonalność to proste dwuoktawowe pianino umożliwiające grę „na żywo” przy pomocy klawiatury lub myszki. Ostatnia pozycja to „piano roll”, czyli kontrolka umożliwiająca ustawianie wybranych dźwięków w wybranym momencie przez dowolnie określoną ilość taktów dla wybranego instrumentu.

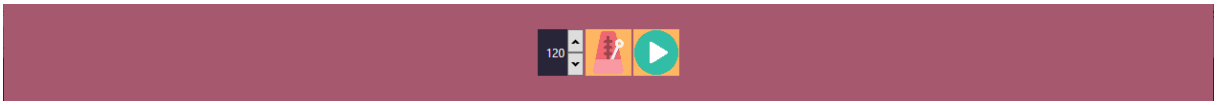
Program jest oparty o bibliotekę nAudio, która daje dużą kontrolę nad plikami dźwiękowymi i udostępnia wiele narzędzi do pracy z nimi. Dodatkowo odtwarzanie dźwięków odbywa się synchroniczne, co zostało zrealizowane przy pomocy wątków i usprawnione oraz zabezpieczone przez zastosowanie Mutex.

3. Specyfikacja zewnętrzna



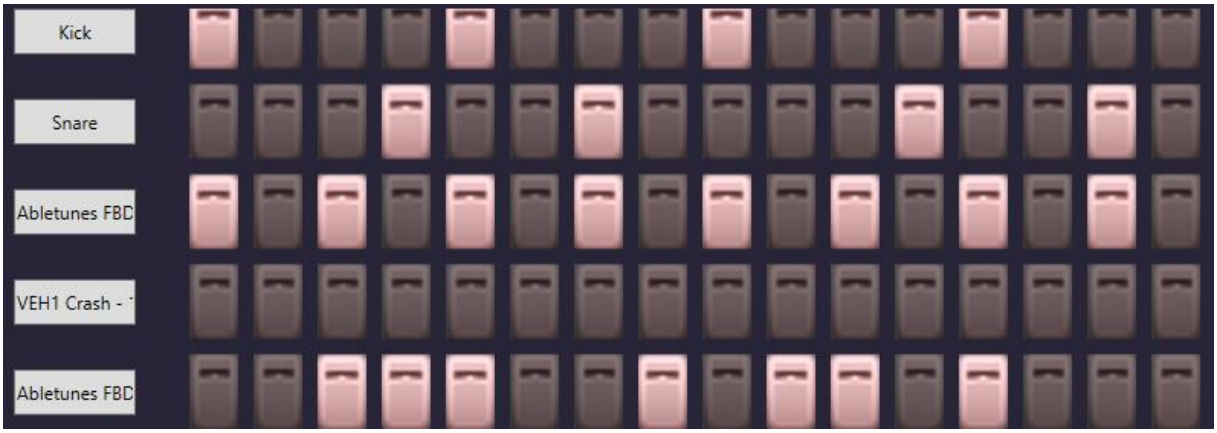
Rys. 1. 1 Początkowy widok programu

Na samej górze jest dostępne proste menu, w którym znajdują się po kolei: kontrolka umożliwiająca ustawienie tempa (uderzenia na minutę), przycisk włączający metronom, przycisk włączający sekwencer i piano roll.



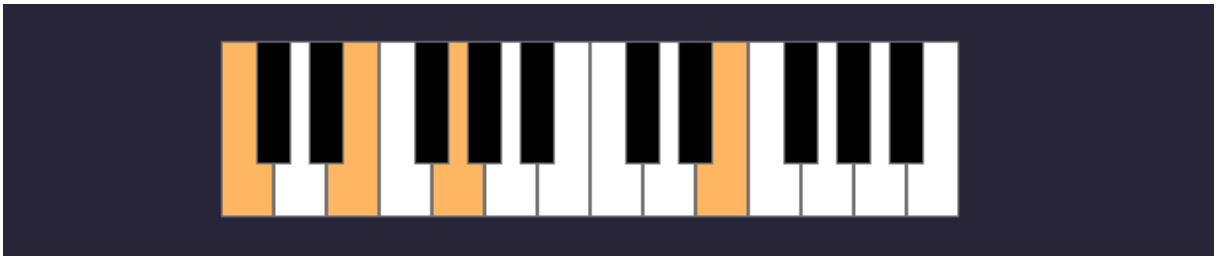
Rys. 1. 2 Menu

Sekwencer składa się z przewijalnej scrollem sekcji umożliwiającej wybieranie, w którym momencie ma być grany jaki dźwięk. Po lewej stronie znajdują się przyciski do wybierania pliku do odtwarzania. Po prawej znajduje się małe menu umożliwiające dodawanie/usuwanie dźwięków oraz analogiczne operacje z ilością taktów.



Rys. 1. 3 Przykładowa konfiguracja sekwencera

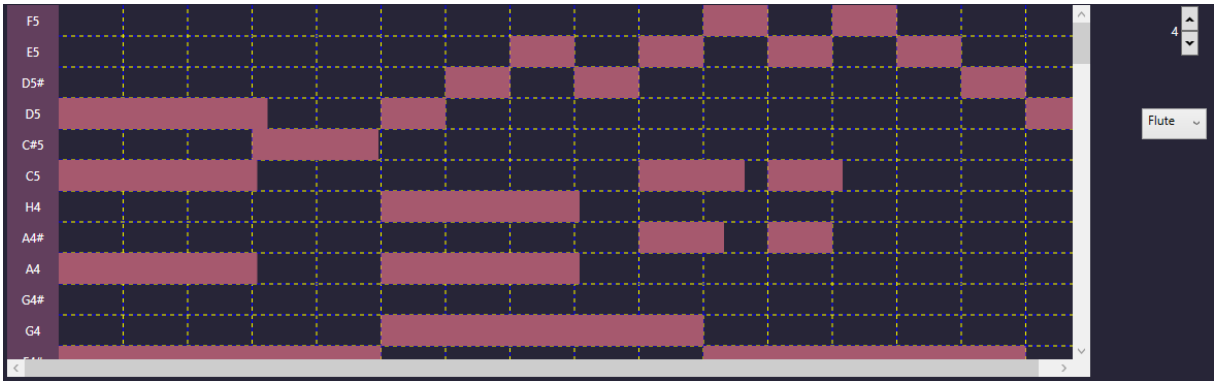
Pianino umożliwia użytkownikowi granie „na żywo” poprzez naciskanie odpowiednich klawiszy klawiatury (konkretne przyciski są przypisane odpowiednim dźwiękom) lub naciskanie lewym przyciskiem myszy narysowanych klawiszy pianina. Pianino nie posiada wyboru odtwarzanych dźwięków oraz umożliwia granie w zasięgu dwóch oktaw. Nie ma limitu na ilość równocześnie naciskanych klawiszy, jednak specyfika urządzenia jakim jest klawiatura nakłada je z góry i zależy ono od modelu. Dźwięk gra tak długo, dopóki naciśniety jest dany klawisz, dodatkowo aktywne przyciski są podświetlane.



Rys. 1. 4 Przykład używania pianina

Piano roll umożliwia rysowanie dźwięków. Rzędy odpowiadają kolejnym dźwiękom, kolumny momentowi (pierwotnie są pokazane 4 takty składające się z 4 potencjalnych ćwierćnut) odtwarzania, a szerokość prostokąta symbolizuje czas trwania dźwięku. Użytkownik ma możliwość rysowania kolejnych nut przy pomocy PPM, usuwania przy pomocy LPM, przesuwania przy pomocy ŚPM oraz zmieniania długości przy pomocy kombinacji LCTRL i używania kółka myszy (scroll w górę wydłuża nutę, w dół skraca). Nie

ma górnego limitu szerokości. Po prawej stronie znajduje się kontrolka do ustawiania liczby taktów oraz lista do wyboru instrumentu (pakiet dźwięków, który ma być odtwarzany). Kolejne nuty rysują się tej samej długości, jaka była poprzednia ustawiona i są wyrównywane do kolumn (niezależnie, gdzie klikniemy w obrębie danego pola, nuta zostanie ustawiona na jego początku). Próba dodania nuty do zajętego pola nie skutkuje dodaniem duplikatu, lecz ustawieniem aktualnej długości rysowania nut na długość naciśniętej nuty.



Rys. 1. 5 Przykładowa konfiguracja piano roll (4 takty i flet jako instrument)

4. Specyfikacja Wewnętrzna

4.1 Znaczenie klas i istotne własności, zależności

MenuButton – odpowiada za kontrolowanie stanu przycisku ***play*** w menu. Posiada pole informujące czy przycisk jest aktywny i metody operujące na nim.

Metronome – odzwierciedla metronom. Posiada pola i metody odpowiedzialne za **ustawianie/informowanie** o **BPM** oraz **licznik uderzeń**, służący do **synchronizacji** kontrolek odpowiadających za granie **dźwięków**. Dziedziczy po klasie ***MenuButton*** (w menu znajduje się przycisk odpowiedzialny za uruchomianie metronomu).

Sequencer – implementuje kontrolkę **sekwencer**. Posiada pola przechowujące ilość taktów, dźwięków, ścieżkę do folderu z plikami audio oraz 3 słowniki: ***notes*** zawiera informacje, w którym momencie zagrać konkretny dźwięk, ***sounds*** posiada nazwy używanych plików audio a ***mutexes*** przechowuje mutex'y dla każdego z granych dźwięków. Implementuje metody odpowiadające za **dodawanie/usuwanie nut** i **dźwięków**.

PianoKeyboard – odpowiada za kontrolkę **pianina**. Posiada 2 słowniki: ***notes*** powiązuje konkretne przyciski klawiatury z odpowiednimi nazwami dźwięków, ***devices*** posiada obiekty klasy ***WaveOutEvent*** (zaczerpnięte z biblioteki ***nAudio)***, które odpowiadają za granie dźwięków. Istnieją również 2 HashSet'y: ***blackKeys*** zawierający reprezentacje przycisków klawiatury, które grają odpowiadają za granie dźwięków chromatyczne (czarne klawisze na pianinie), oraz ***arePlaying*** zawierający informacje, które dźwięki są aktualnie grane. Klasa posiada metody odpowiedzialne za granie dźwięków przy pomocy widniejącego w interfejsie pianina dwuoktawowego.

RollKey – odpowiada za reprezentacje rysowanej **nuty** w postaci prostokąta na siatce kontrolki ***Piano Roll***. Posiada m.in. pola z obiektem klasy ***Rectangle*** (biblioteka ***System.Windows.Shapes***), jego ***długość*** oraz ***współrzędne*** (wiersz oraz kolumna w **gridzie**). Posiada metody odpowiadające **modyfikowanie** lub zwracanie powyższych pól.

PianoRoll – implementuje kontrolę **piano roll**. Odpowiada za **interakcję użytkownika** z rysowanymi nutami (prostokątami). Nuty są rysowane na **Canvie** i jednocześnie umieszczane w **Gridzie**. Klasa posiada pola przechowujące **ilość taktów**, jakiej **długości** rysować nuty oraz **ilość** narysowanych obecnie nut. W dodatku zawiera 2 słowniki: ***notes*** przechowujący obiekty klasy ***RollKey*** oraz słownik (***graph***) słowników z informacją o **położeniu nuty w czasie** wraz z jej **długością**. Dodatkowo Klasa posiada metody odpowiadające za r**ysowanie**, **usuwanie**, **przemieszczanie** oraz **zmianę długości nut**.

IRollPlayer – **Interfejs** odpowiadający za ustawianie nazw dźwięków na ***Gridzie*** oraz za powiązywanie kolejnych rzędów kontrolki ***piano roll*** z odpowiednimi **dźwiękami**. W Programie zostało zawartych 5 klas **dziedziczących**: PresetChoir, PresetPiano, PresetFlute, PresetOrgans oraz PresetSaxophone. Wszystkie odpowiadają za konkretny instrument a rozszerzanie programu o kolejne ***„Presety”*** jest bardzo proste dzięki zastosowaniu wzorca projektowego ***strategia***.

Context – Obsługuje główny ***mechanizm wzorca strategia***. Posiada pole typu ***IRollPlayer*** oraz implementuje metodę umożliwiającą **wybór strategii** (**instrumentu**).

Presets – **Wrapper** do przechowywania **instrumentów** (obiektów ***IRollPlayer***) używanych przez **piano roll**.

4.2 Wykorzystane struktury danych

Program wykorzystuje **listy**, **słowniki** oraz **HashSety**. W niektórych miejscach zostały użyte wersje **wątkowo-bezpieczne** (z przedrostkiem ***concurrent***). Zastosowane struktury są optymalne dla oczekiwanych zestawów danych i dają zadowalające rezultaty prędkości wykonywanych operacji oraz zajmowanego rozmiaru pamięci.

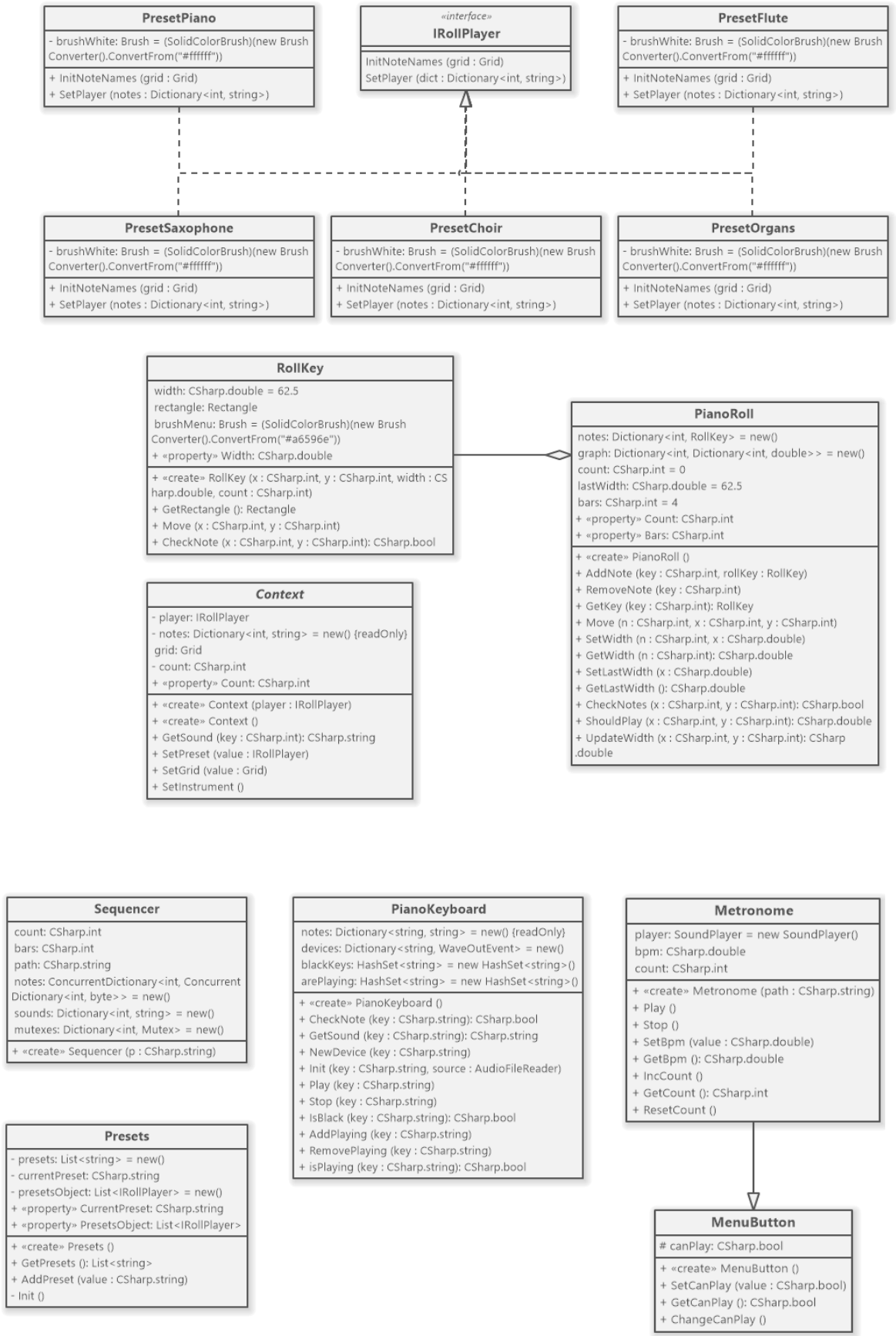
4.3 Zastosowane techniki obiektowe

- ***Abstrakcja*** (zastosowanie ***interfejsu*** *IRollPlayer*)
- ***Dziedziczenie*** (Metronome ***dziedziczy*** po *MenuButton*)
- ***Polimorfizm*** (klasy implementujące *IRollPlayer*)
- ***Asocjacja***, ***agregacja***, ***kompozycja*** i ***zależność***
- ***Enkapsulacja***

4.4 Zastosowane zagadnienia z laboratorium

- ***Wątki*** (**przeszukiwanie** struktur oraz **granie** dźwięków, odbywa się w wielu wątkach)
- ***Mutex*** (zabezpieczenie **czytania** pliku Audio z jednoczesnym **odtwarzaniem** tego dźwięku)
- ***Regex*** (w kontrolce menu odpowiadającej za podawanie ***BPM*** można podawać tylko **cyfry**)
- ***System.IO*** (odpowiednik **filesystem**, zastosowany do określania dostępnych **instrumentów)**

4.5 Diagram Klas



Rys. 1. 6 Diagram klas