

**AKADEMIA NAUK STOSOWANYCH
W NOWYM SĄCZU**

WYDZIAŁ NAUK INŻYNIERYJNYCH

PRACA DYPLOMOWA

**Edytor zdjęć na platformę desktopową w .NET MAUI z
wykorzystaniem SkiaSharp**

Autor: Łukasz Nowak

Kierunek: Informatyka

Nr albumu: 32682

Promotor: dr Grzegorz Litawa

Akceptacja promotora:
data i podpis

NOWY SĄCZ 2025

Spis treści

1	Wstęp	5
2	Cel pracy	6
3	Ogólny przegląd technologii	8
3.1	Charakterystyka wybranych technologii	8
3.1.1	.NET MAUI	8
3.1.2	SkiaSharp	9
3.1.3	C# i .NET	9
3.2	Wnioski z analizy technologii	9
4	Projekt rozwiązania oraz wybór technologii	10
4.1	Założenia funkcjonalne	10
4.2	Architektura aplikacji	10
4.3	Wybór technologii	11
5	Implementacja	12
6	Prezentacja rozwiązania	13
7	Podsumowanie	14

1 Wstęp

W dzisiejszych czasach aplikacje do edycji zdjęć odgrywają ważną rolę zarówno w codziennym użytkowaniu, jak i w profesjonalnych zastosowaniach. Celem pracy jest stworzenie desktopowej aplikacji umożliwiającej podstawową edycję grafiki w oparciu o technologię .NET MAUI oraz bibliotekę SkiaSharp.

2 Cel pracy

Współczesne potrzeby użytkowników w zakresie edycji obrazów stale rosną, zarówno w kontekście amatorskiego przetwarzania zdjęć, jak i profesjonalnego przygotowania materiałów graficznych. Celem niniejszej pracy inżynierskiej jest stworzenie desktopowej aplikacji umożliwiającej wygodną edycję zdjęć oraz podstawowe operacje graficzne przy wykorzystaniu technologii .NET MAUI oraz biblioteki graficznej SkiaSharp.

Projektowana aplikacja ma umożliwiać użytkownikom pracę w środowisku warstwowym, co znacząco zwiększa elastyczność i komfort edytowania obrazów. Każda operacja edycyjna, taka jak rysowanie, zmiana koloru czy dodawanie efektów, będzie realizowana na wybranej warstwie, pozwalając użytkownikowi na późniejsze modyfikacje bez konieczności ingerencji w całą grafikę.

W ramach realizacji założeń projektowych planuje się osiągnięcie następujących celów funkcjonalnych:

- **Zaprojektowanie i implementacja intuicyjnego interfejsu użytkownika** dostosowanego do standardów aplikacji desktopowych, zapewniającego szybki dostęp do narzędzi edycyjnych oraz wygodne zarządzanie projektem graficznym.
- **Stworzenie systemu zarządzania warstwami**, umożliwiającego dodawanie, usuwanie, wybieranie oraz ukrywanie warstw, a także zapamiętywanie ich parametrów w projekcie.
- **Implementacja podstawowych narzędzi graficznych** takich jak:
 - Pędzel o regulowanej grubości i kolorze,
 - Gumka umożliwiająca usuwanie fragmentów obrazu,
 - Narzędzie do przycinania obszaru obrazu,
 - Narzędzie wypełniania wybranego obszaru kolorem.
- **Obsługa operacji na plikach graficznych**, w tym:
 - Zapis gotowego obrazu w formatach PNG, JPG, BMP,
 - Otwieranie istniejących plików graficznych do edycji,
 - Zapis i odczyt projektu w dedykowanym formacie (np. plik JSON lub XML) z zachowaniem informacji o warstwach.
- **Integracja z systemem plików** w celu umożliwienia użytkownikowi swobodnego zarządzania projektami i obrazami w środowisku Windows.

Realizacja powyższych celów pozwoli na stworzenie lekkiej, funkcjonalnej aplikacji graficznej, która znajdzie zastosowanie zarówno w codziennym użytkowaniu (np. edycja zdjęć na potrzeby mediów społecznościowych), jak i w prostych projektach graficznych wymagających pracy na wielu warstwach.

W szerszej perspektywie, projekt ten umożliwi również wykazanie praktycznych umiejętności w zakresie programowania aplikacji graficznych oraz obsługi technologii wieloplatformowych. W przyszłości aplikacja może zostać rozbudowana o bardziej zaawansowane funkcje, takie jak nakładanie filtrów graficznych, zaawansowane operacje zaznaczania czy integrację z popularnymi formatami grafiki wektorowej (np. SVG).

3 Ogólny przegląd technologii

Przy projektowaniu aplikacji desktopowej do edycji grafiki konieczny był wybór odpowiednich technologii, które zapewnią wysoką wydajność, elastyczność oraz możliwość łatwej rozbudowy w przyszłości. W początkowej fazie analizy rozważono kilka popularnych rozwiązań stosowanych w projektach graficznych:

- **Windows Presentation Foundation (WPF)** – popularny framework Microsoftu do budowy aplikacji desktopowych dla Windows, oferujący rozbudowane wsparcie dla tworzenia graficznych interfejsów użytkownika i renderowania grafiki wektorowej.
- **WinForms** – starsze rozwiązanie Microsoftu, nadal wykorzystywane w wielu projektach korporacyjnych, charakteryzujące się prostotą implementacji, ale ograniczonymi możliwościami w zakresie grafiki nowoczesnej.
- **GTK#** – zestaw narzędzi graficznych stosowany głównie w systemach Linux, umożliwiający tworzenie aplikacji wieloplatformowych.
- **.NET MAUI (Multi-platform App UI)** – nowe środowisko programistyczne od Microsoftu, nastawione na tworzenie aplikacji wieloplatformowych przy użyciu jednego kodu źródłowego, następca technologii Xamarin.Forms.
- **Qt** – bardzo rozbudowany framework wykorzystywany w aplikacjach cross-platform, głównie w środowiskach C++.

Po przeprowadzeniu analizy porównawczej pod kątem nowoczesności, możliwości graficznych, wsparcia dla nowoczesnych systemów Windows oraz łatwości integracji z C#, zdecydowano się na wybór technologii .NET MAUI w połączeniu z biblioteką graficzną SkiaSharp.

3.1 Charakterystyka wybranych technologii

3.1.1 .NET MAUI

.NET MAUI umożliwia budowanie natywnych aplikacji dla różnych platform z wykorzystaniem wspólnego kodu C#. Zapewnia nowoczesne mechanizmy tworzenia interfejsów użytkownika, wsparcie dla wzorców MVU (Model-View-Update) oraz rozbudowane możliwości adaptacji układu aplikacji do różnych rozdzielczości ekranów.

Dodatkowo MAUI integruje się płynnie z bibliotekami graficznymi oraz umożliwia wykorzystanie natywnych zasobów urządzenia, co znacząco przyspiesza rozwój aplikacji.

3.1.2 SkiaSharp

SkiaSharp to potężna biblioteka graficzna pozwalająca na niskopoziomowe rysowanie bit-map, kształtów, ścieżek oraz tekstu. Biblioteka ta została zaadaptowana na platformę .NET i zapewnia wsparcie dla wydajnych operacji graficznych, co ma kluczowe znaczenie w aplikacji edytującej obrazy w czasie rzeczywistym.

SkiaSharp charakteryzuje się wysoką wydajnością, obsługą renderowania antyaliasingowego oraz możliwością stosowania filtrów i przekształceń grafiki, co czyni ją idealnym wyborem dla edytora grafiki 2D.

3.1.3 C# i .NET

C# jako język programowania został wybrany ze względu na swoją dojrzałość, szerokie wsparcie w środowisku Microsoft oraz nowoczesne funkcje ułatwiające pracę nad dużymi projektami (takie jak LINQ, asynchroniczność, bezpieczeństwo typów). Platforma .NET zapewnia natomiast bogaty ekosystem bibliotek oraz narzędzi, co przyspiesza proces tworzenia aplikacji.

3.2 Wnioski z analizy technologii

Wybór .NET MAUI i SkiaSharp umożliwia realizację założonych celów projektu, zapewniając nowoczesne środowisko programistyczne, wysoką wydajność operacji graficznych oraz łatwość późniejszej rozbudowy aplikacji. Dzięki temu możliwe będzie stworzenie aplikacji spełniającej zarówno wymagania funkcjonalne, jak i jakościowe postawione na początku prac projektowych.

4 Projekt rozwiązania oraz wybór technologii

Projekt rozwiązania zakłada stworzenie desktopowej aplikacji umożliwiającej edycję zdjęć przy pomocy intuicyjnego interfejsu użytkownika oraz systemu pracy na warstwach. W celu zapewnienia wysokiej wydajności operacji graficznych oraz komfortu użytkowania, projekt został oparty o technologie .NET MAUI oraz SkiaSharp.

4.1 Założenia funkcjonalne

Aplikacja ma umożliwiać użytkownikowi wykonywanie następujących operacji:

- Tworzenie nowego projektu graficznego o określonym rozmiarze.
- Dodawanie, usuwanie oraz zarządzanie warstwami obrazu.
- Rysowanie na aktywnej warstwie za pomocą pędzla o regulowanej grubości i kolorze.
- Korzystanie z gumki w celu usuwania niepożądanych fragmentów grafiki.
- Zmiana kolorów przy pomocy suwaków RGB lub wyboru z predefiniowanych kolorów.
- Zapisywanie projektu w formacie własnym, umożliwiającym późniejsze wznowienie pracy (z zachowaniem struktury warstw).
- Eksport gotowego obrazu do popularnych formatów graficznych (PNG, JPG, BMP).
- Otwieranie istniejących plików graficznych do edycji.
- Możliwość przycinania obrazu do wybranego fragmentu.

4.2 Architektura aplikacji

Aplikacja została zaprojektowana w sposób modułowy, co pozwala na łatwe rozbudowywanie oraz testowanie poszczególnych komponentów. Wyróżniono następujące główne moduły:

- **Moduł Interfejsu Użytkownika (UI)** — odpowiadający za wyświetlanie płótna roboczego, paneli narzędziowych oraz zarządzanie interakcjami użytkownika.
- **Moduł Warstw** — zarządzający strukturą warstw obrazu, umożliwiający ich tworzenie, edytowanie i usuwanie.
- **Moduł Operacji Graficznych** — odpowiedzialny za rysowanie kształtów, obsługę pędzla, gumki oraz wypełnień na odpowiednich warstwach.
- **Moduł Operacji na Plikach** — obsługujący zapis i odczyt obrazów oraz projektów graficznych z warstwami.

Schemat działania aplikacji opiera się na wyborze aktywnej warstwy, na której wykonywane są wszystkie operacje rysunkowe i edycyjne. Zmiana warstwy powoduje aktualizację kontekstu rysowania, a operacje takie jak zapis czy eksport uwzględniają wszystkie warstwy w kolejności ich położenia.

4.3 Wybór technologii

Podczas projektowania aplikacji kluczowe znaczenie miały następujące kryteria wyboru technologii:

- Wydajność operacji graficznych — konieczność zapewnienia płynności rysowania nawet przy większych rozdzielczościach obrazów.
- Łatwość budowy intuicyjnego interfejsu użytkownika — aby zminimalizować czas potrzebny na naukę obsługi aplikacji.
- Możliwość pracy wieloplatformowej w przyszłości — chociaż głównym celem jest aplikacja desktopowa na Windows, potencjalna rozbudowa o wersję mobilną była brana pod uwagę.
- Stabilność i wsparcie społeczności — wybór technologii aktywnie rozwijanych i dobrze udokumentowanych.

Na podstawie powyższych kryteriów zdecydowano się na następujące technologie:

- **.NET MAUI** — jako nowoczesny framework umożliwiający budowę nowoczesnych aplikacji desktopowych, a w przyszłości także mobilnych.
- **SkiaSharp** — jako szybki i elastyczny silnik grafiki 2D, doskonale nadający się do implementacji operacji rysunkowych i pracy na bitmapach.
- **C# i .NET** — jako główny język programowania oraz środowisko wykonawcze aplikacji.
- **System.IO oraz JSON** — do zarządzania plikami graficznymi oraz formatami projektów z warstwami.

Wybór tych narzędzi pozwala na realizację wszystkich założeń projektowych oraz umożliwia elastyczne rozwijanie aplikacji w przyszłości.

5 Implementacja

Implementacja obejmowała stworzenie szkieletu aplikacji, systemu zarządzania warstwami, funkcji rysowania pędzlem, obsługi zapisu i odczytu obrazów oraz interfejsu użytkownika z intuicyjną obsługą kolorów i narzędzi.

6 Prezentacja rozwiązania

Gotowa aplikacja umożliwia użytkownikowi tworzenie nowych projektów graficznych, edycję obrazów przy pomocy różnych narzędzi, zapisywanie plików w popularnych formatach oraz pracę na wielu warstwach.

7 Podsumowanie

Stworzona aplikacja spełnia założone cele, oferując intuicyjny interfejs oraz funkcjonalność edytora grafiki 2D opartego na systemie warstw. W przyszłości możliwe jest rozszerzenie projektu o bardziej zaawansowane funkcje takie jak filtry czy obsługa formatów wektorowych.