



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: Programowanie

Hubert Michna
Nr albumu studenta w67259

"Samochód"

Prowadzący: mgr inż. Ewa Żesławska

Praca projektowa programowanie obiektowe C#

Rzeszów 2023-2024

Spis treści

1	Opis założeń projektu	4
1.1	Cele projektu	4
1.2	Wymagania funkcjonalne	4
1.3	Wymagania нефункционалне	4
2	Opis struktury projektu	6
2.1	Diagram Klas	6
2.1.1	Klasa abstrakcyjna: 'Samochód'	6
2.1.2	Klasa abstrakcyjna: 'Rodzaj_Napędu'	7
2.1.3	Klasa abstrakcyjna: 'Skrzynia_Biegów'	8
2.2	Struktura plików .txt	8
2.2.1	Przykład	8
2.3	Struktura bazy danych	8
2.4	Struktura plików .csv	9
2.4.1	Przykład	9
2.5	Opis techniczny projektu	10
2.5.1	Wymagania sprzętowe	10
3	Harmonogram realizacji projektu	11
3.1	Diagram Gantta	11
3.2	Repozytorium	11
4	Prezentacja warstwy użytkowej projektu	12
4.1	Zakładka 'Symulacja'	12
4.1.1	Zakładka 'Symulacja', Lewa Kolumna	12
4.1.2	Zakładka 'Symulacja', Środkowa Kolumna	13
4.1.3	Zakładka 'Symulacja', Prawa Kolumna	13
4.2	Zakładka 'Wybór Samochodu'	14
4.2.1	Zakładka 'Wybór Samochodu', Lewa Kolumna	14
4.2.2	Zakładka 'Wybór Samochodu', Prawa Kolumna	15
5	Podsumowanie	16
	Bibliografia	17
	Spis rysunków	18

Rozdział 1

Opis założeń projektu

1.1 Cele projektu

Ten projekt powstał w celu umożliwienia przeprowadzenia symulacji jazdy samochodów (osóbek, dostawczaków i TIR'ów).

Symulacja pozwala na przyspieszanie, hamowanie, hamowanie pasywne, zmianę biegów oraz uzupełnianie oleju i paliwa. Jak również możliwość obserwowania parametrów takich jak prędkość, ilość obrotów na minutę, ilość oleju czy ilość paliwa (lub pozostały czas jazdy).

Samochody dostępne w symulacji są wczytywane z plików .txt oraz .csv, bazy danych. Projekt pozwala na dodawanie nowych samochodów i zapisywanie ich w tych formatach, tak aby mogły być wykorzystane w przyszłych symulacjach.

Wynikiem tego projektu jest program komputerowy który pozwala na wspomnianą symulację.

1.2 Wymagania funkcjonalne

Program pozwala na wybór samochodu za pomocą listy, która zawiera informacje o parametrach samochodu. Podczas wybierania samochodu istnieje możliwość edycji jego parametrów.

Po wybraniu samochodu można przeprowadzić symulację, poprzez włączanie i wyłączanie silnika, przyspieszanie, hamowanie, hamowanie pasywne i zmiany biegów, oraz uzupełnianie paliwa i oleju.

Podczas trzymania przycisku przyspieszania zwiększa się prędkość pojazdu oraz ilość obrotów na minutę (RPM).

Podczas trzymania przycisku hamowania a także hamowanie pasywnego, zmniejsza się prędkość pojazdu oraz ilość obrotów na minutę.

Biegi można zmieniać za pomocą grupy przycisków z podpisanymi numerami biegów.

Wartości parametrów takich jak prędkość, RPM, ilość pozostałego paliwa i oleju są wyświetlane na ekranie. Ich wartości są modyfikowane co określony moment czasu podczas trwania symulacji.

1.3 Wymagania niefunkcjonalne

Program posiada intuicyjny interfejs graficzny, wykonany według nowoczesnych standardów, który pozwala na przeprowadzanie symulacji oraz wprowadzanie danych przez użytkownika.

Ten program jest przeznaczony na komputery osobiste z systemem operacyjnym Windows.

Dane o samochodach mogą być zarówno zapisywane i odczytywane z różnych formatów plików (np. pliki .txt, .csv lub jako rekordy w bazie danych.) co umożliwia bezpieczeństwo danych, poprzez danie możliwości tworzenia kopii zapasowych tych plików we własnym zakresie. Co pozwala też na wymianę plików pomiędzy urządzeniami i przeprowadzanie symulacji na innych komputerach.

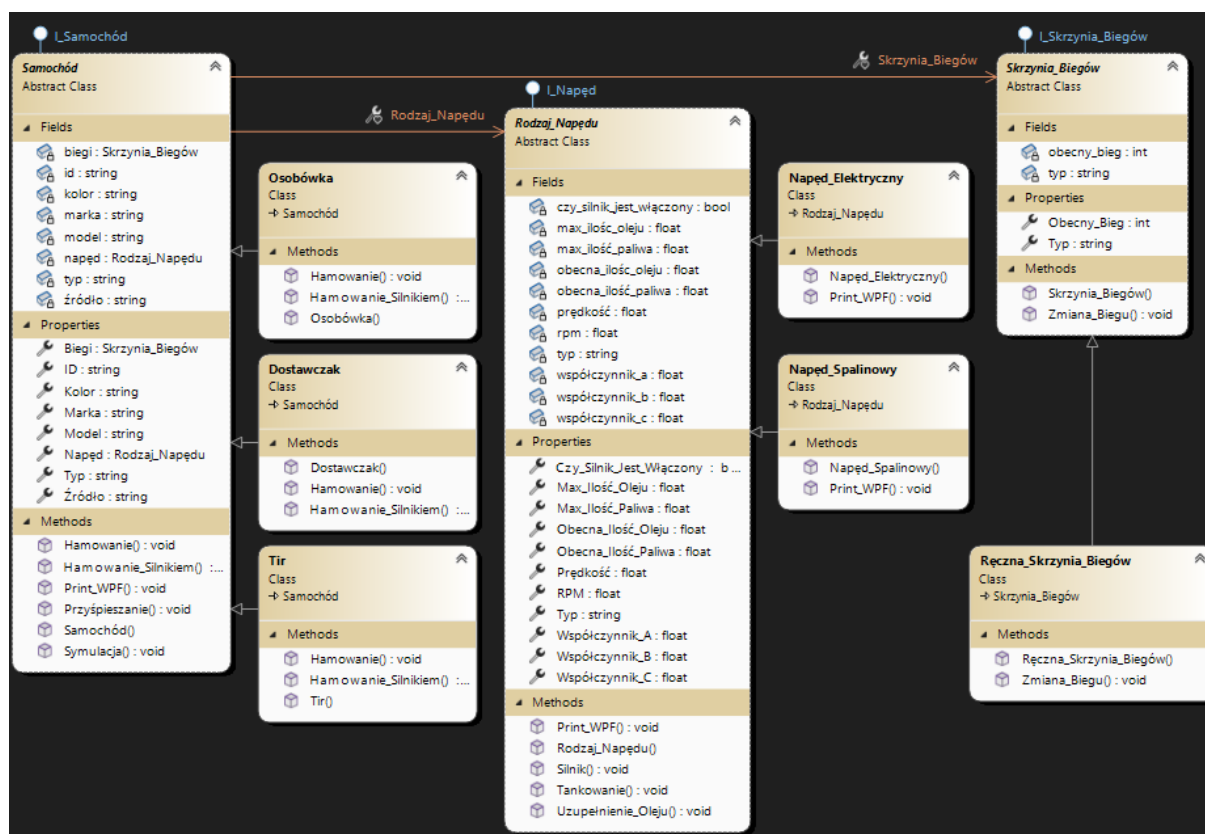
Program wykorzystuje bazę danych SQLite, przez co nie ma potrzeby na korzystanie z serwera baz danych. Z tego wynika że nie ma kosztów utrzymania takiego serwera oraz to że aplikacja nie potrzebuje dostępu do Internetu do poprawnego działania.

Rozdział 2

Opis struktury projektu

W tym rozdziale mają pojawić się informacje odnośnie zaprojektowanej struktury oraz jej opis wraz z opisem technicznym. Należy umieścić informacje odnośnie wykorzystywanego języka, narzędzi oraz minimalnych wymagań sprzętowych. Opisać zarządzanie danymi oraz BD, umieścić informacje odnośnie zaprojektowanej hierarchii klas wraz z krótkim opisem najważniejszych metod.

2.1 Diagram Klas



Rysunek 2.1: Diagram Klas

2.1.1 Klasa abstrakcyjna: 'Samochód'

Ta klasa abstrakcyjna zawiera informacje o samochodach, najważniejsze metody związane z symulacją działania samochodu (np. przyspieszanie czy hamowanie). Służy w celu pogrupowania poszcze-

gólnych typów samochodów w jedną kolekcję, która może być potem z łatwością wyświetlana jako zawartość DataGridView (tabelka) w zakładce "Wybór Samochodu". Jej najważniejsze metody to:

- **Print_WPF(MainWindow GUI):** Ta metoda aktualizuje stan napisów wyświetlanych w lewej kolumnie, na zakładce "Symulacja"
- **Symulacja(MainWindow GUI):** Ta metoda wykonuje symulację opartą na właściwościach danego samochodu. Ta metoda jest wywoływana co określony odstęp czasowy.
- **Przyśpieszanie():** Zmienia prędkość pojazdu oraz ilość RPM w zależności od współczynników, odpowiednio współczynnika A i współczynnika B, kiedy wciśnięty jest przycisk przyśpieszania. Jest wywoływana przez metodę Samochód.Symulacja(MainWindow GUI).
- **Hamowanie():** Zmniejsza prędkość oraz RPM danego pojazdu, kiedy wciśnięty jest przycisk hamowania. Jest wywoływana przez metodę Samochód.Symulacja(MainWindow GUI).
- **Hamowanie_Pasywne():** Zmniejsza prędkość oraz RPM danego pojazdu (w mniejszym stopniu niż metoda Samochód.Hamowanie()), kiedy nie jest wciśnięty przycisk przyśpieszania ani hamowania. Jest wywoływana przez metodę Samochód.Symulacja(MainWindow GUI).

Klasa pochodna: 'Osobówka'

Klasa "Osobówka" jest klasą pochodną klasy "Samochód", zawiera ona informacje o dostawczakach. Są tutaj zawarte rozszerzenia metod z klasy Samochód.

Klasa pochodna: 'Dostawczak'

Klasa "Dostawcza" jest klasą pochodną klasy "Samochód", zawiera ona informacje o dostawczakach. Są tutaj zawarte rozszerzenia metod z klasy Samochód.

Klasa pochodna: 'Tir'

Klasa "Tir" jest klasą pochodną klasy "Samochód", zawiera ona informacje o samochodach ciężarowych. Są tutaj zawarte rozszerzenia metod z klasy Samochód.

2.1.2 Klasa abstrakcyjna: 'Rodzaj_Napędu'

Ta klasa abstrakcyjna zawiera informacje o napędzie samochodu, zawiera informacje między innymi o prędkości, ilości obrotów na minutę czy ilości paliwa i oleju. Tutaj są zawarte parametry, które mają wpływ na przebieg symulacji dla każdego samochodu. Jej najważniejsze metody to:

- **Silnik(MainWindow GUI):** Ta metoda włącza silnik samochodu, ustawia również odpowiednie wartości dla prędkości i RPM. Wywoływana po wciśnięciu przycisku włączania/wyłączania silnika.
- **Tankowanie(MainWindow GUI):** Uzupełnia ilość paliwa jeśli samochód ma wyłączony silnik i nie porusza się. Wywoływana poprzez wciśnięcie przycisku tankowania.
- **Uzupełnienie_Oleju(MainWindow GUI):** Uzupełnia ilość oleju jeśli samochód ma wyłączony silnik i nie porusza się. Wywoływana poprzez wciśnięcie przycisku uzupełniania oleju.

Klasa pochodna: 'Napęd_Spalinowy'

Klasa "Napęd_Spalinowy" jest klasą pochodną klasy "Rodzaj_Napędu", zawiera ona informacje o napędach spalinowych dla samochodów.

Klasa pochodna: 'Napęd_Elektryczny'

Klasa "Napęd_Spalinowy" jest klasą pochodną klasy "Rodzaj_Napędu", zawiera ona informacje o napędach elektrycznych dla samochodów.

2.1.3 Klasa abstrakcyjna: 'Skrzynia_Biegów'

Ta klasa abstrakcyjna zawiera informacje o skrzyni biegów samochodu, zawiera informacje o wybranym biegu oraz podstawową logikę związaną z wyborem biegu. Jej najważniejsza metoda to:

- Zmiana_Biegu(): Obsługuje zmianę biegów i podstawową związaną z nią logikę. Wywoływana poprzez wciskanie przycisków ze zmianą biegów.

Klasa pochodna: 'Ręczna_Skrzynia_Biegów'

Klasa "Ręczna_Skrzynia_Biegów" jest klasą pochodną klasy "Skrzynia_Biegów", zawiera ona informacje o ręcznej skrzyni biegów oraz związaną z nią logikę zmiany biegów, która jest rozszerzeniem tej z klasy bazowej.

2.2 Struktura plików .txt

Pliki tekstowe zawierają dane w oddzielnych liniach, ich kolejność jest dowolna, ale muszą mieć określoną strukturę. Każda linijka musi się zaczynać od nazwy właściwości, zakończoną dwukropkiem i spacją, następnie może być podawana wartość, cała linijka musi być zakończona znakiem średnika (";"). Separatorem dziesiętnym jest przecinek (",").

2.2.1 Przykład

Zawartość przykładowego pliku .txt:

```
Typ: Osobówka;  
Model: Tucson;  
Marka: Hyundai;  
Rodzaj Napędu: Spalinowy;  
Skrzynia Biegów: Ręczna;  
Kolor: Srebrny;  
Współczynnik A: 0,7;  
Współczynnik B: 83,5;  
Współczynnik C: 0,0027;
```

Aby pliki ".txt" mogły zostać obsłużone przez program, muszą się znajdować w odpowiedniej lokalizacji, domyślnie: "../..../Źródła Danych/", względem pliku ".exe".

2.3 Struktura bazy danych

Baza danych SQLite, o nazwie "Projekt_v1.db" zawiera tylko jedną tabelę, ma ona nazwę samochodu. Posiada ona następujące kolumny:

1. ID - Klucz główny, INT
2. Typ - STRING
3. Marka - STRING

4. Model - STRING
5. Kolor - STRING
6. Rodzaj_Napedu - STRING
7. Skrzynia_Biegow - STRING
8. Wspolczynnik_A - STRING
9. Wspolczynnik_B - STRING
10. Wspolczynnik_C - STRING

Aby baza danych SQLite mogła zostać obsługiwana przez program, musi się znajdować w odpowiedniej lokalizacji, domyślnie: "..\\..\\..\\Źródła Danych\\Projekt_v1.db", względem pliku ".exe".

2.4 Struktura plików .csv

Plik "Dane.csv" zawiera dane oddzielone średnikami (";"), plik nie zawiera nagłówka z nazwami kolumn. Separatorem dziesiętnym jest przecinek (",") Zawarte w nim dane muszą być w następującej kolejności:

1. Typ - Wartość: "Osobówka", "Dostawczak" lub "Tir"
2. Marka - Tekst
3. Model - Tekst
4. Kolor - Tekst
5. Rodzaj Napędu - Wartość: "Spalinowy" lub "Elektryczny"
6. Rodzaj Skrzyni Biegów - Zawsze wartość "Ręczna"
7. Współczynnik A - Float
8. Współczynnik B - Float
9. Współczynnik C - Float

Aby plik ".csv" mógł zostać obsługiwany przez program, musi się on znajdować w odpowiedniej lokalizacji, domyślnie: "..\\..\\..\\Źródła Danych\\Dane.csv", względem pliku ".exe".

2.4.1 Przykład

Zawartość przykładowego pliku Dane.csv:

```
Osobówka;Kia;XCeed;Granatowy;Spalinowy;Ręczna;0,68;66,7;0,0024
Osobówka;Skoda;Fabia;Zielony;Spalinowy;Ręczna;0,72;50;0,0018
Osobówka;Toyota;Corolla;Czerwony;Spalinowy;Ręczna;0,8;83,5;0,0025
Osobówka;Dacia;Sandero;Brązowy;Spalinowy;Ręczna;0,72;83,5;0,0019
Osobówka;Audi;Q5;Srebrny;Spalinowy;Ręczna;0,96;108,5;0,0028
Osobówka;Dacia;Spring;Brązowy;Elektryczny;Ręczna;0,5;200;0,0023
Osobówka;Mercedes;EQA;Szary;Elektryczny;Ręczna;0,64;200;0,002
Osobówka;Porsche;Taycan;Czarny;Elektryczny;Ręczna;1,04;260;0,0017
```

2.5 Opis techniczny projektu

Projekt został wykonany w języku C# za pomocą zintegrowanego środowiska programistycznego "Visual Studio" w wersji 2022 z wtyczką "Microsoft.Data.Sqlite", która pozwala na połączenie z bazą danych SQLite.

Baza danych SQLite została przygotowana wykorzystując narzędzie "SQLiteStudio", posiada ono interfejs graficzny, który jest intuicyjny i prosty w obsłudze.

2.5.1 Wymagania sprzętowe

Program został przygotowany z myślą o komputerach osobistych, z systemem operacyjnym Windows 10, który w ustawieniach regionalnych ma przypisany znak przecinka (",") jako separator dziesiętny. Z czego wynika że minimalne wymagania sprzętowe to:

- **System operacyjny:** Windows 10 64bit
- **Procesor:** min 1GHz
- **RAM:** 2GB
- **Miejsce na dysku:** 20GB
- **GPU:** Obsługa DirectX 9 lub nowsza ze sterownikiem WDDM 1.0

Program został przetestowany na komputerze o parametrach:

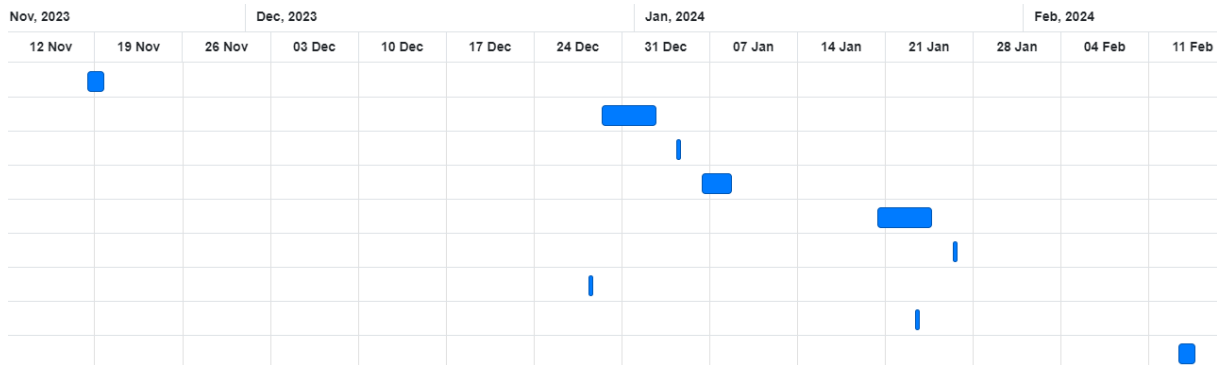
- **System Operacyjny:** Windows 10, wersja: 22H2
- **Procesor:** i7-9700
- **RAM:** 32GB 2667MHz
- **GPU:** RTX 2060 Super

Rozdział 3

Harmonogram realizacji projektu

3.1 Diagram Gantta

Wczesne prace nad projektem rozpoczęły się w połowie listopada 2023 roku. Większość nakładu pracy została zrealizowana na przełomie grudnia i stycznia. Dokumentacja została przygotowana w połowie stycznia.



Rysunek 3.1: Diagram Gantta

3.2 Repozytorium

Poniżej znajduje się link do repozytorium GitHub z plikami projektu:

https://github.com/Hubsik456/Projekt_OOP

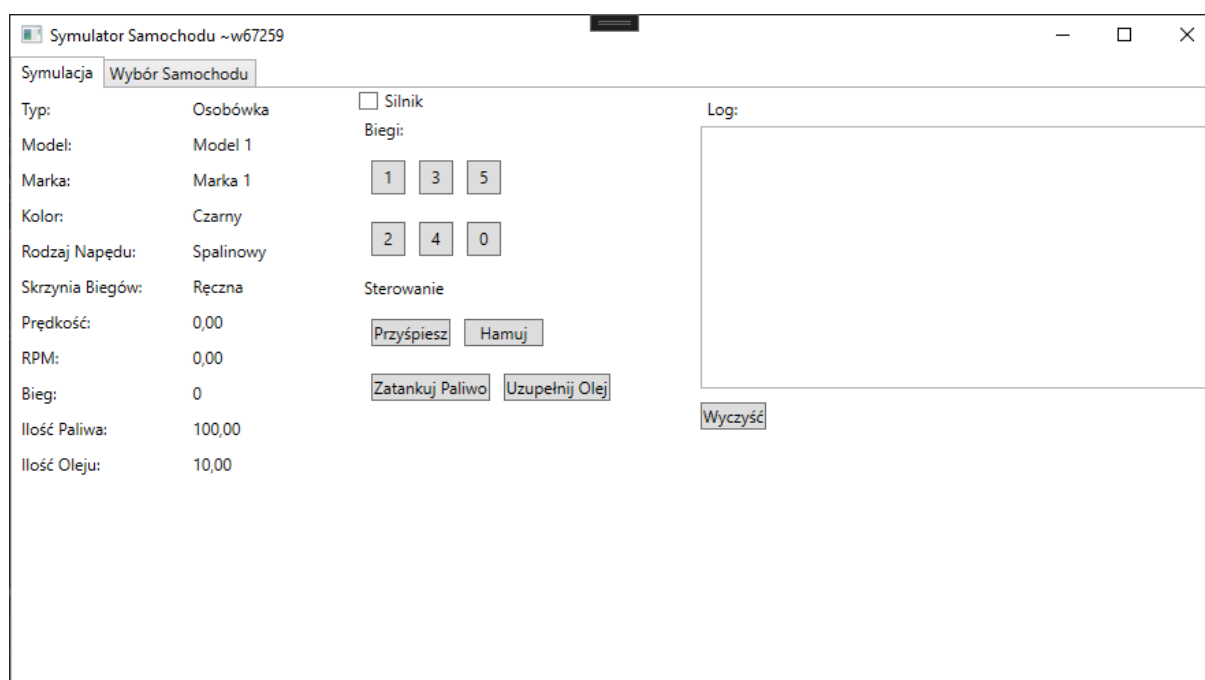
Rozdział 4

Prezentacja warstwy użytkowej projektu

Po uruchomieniu programu jest widoczne jedno, główne okno, które jest podzielone na dwie zakładki "Symulacja" oraz "Wybór Samochodu", które są opisane poniżej.

4.1 Zakładka 'Symulacja'

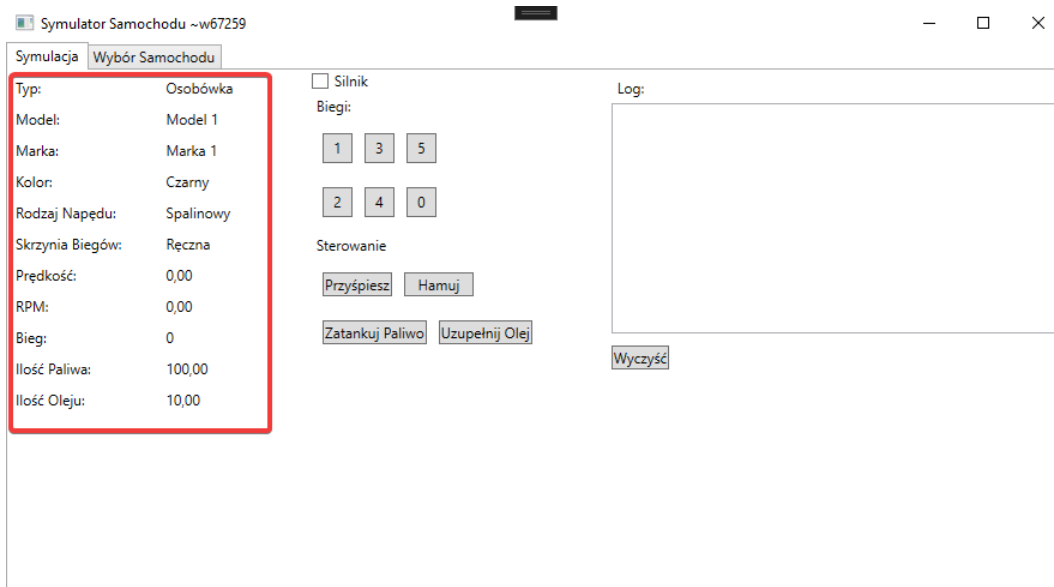
Zakładka "Symulacja" pozwala na sterowanie symulacją samochodu, zmianę biegów, włączanie i wyłączanie silnika, uzupełnianie paliwa czy oleju oraz obserwowanie parametrów samochodu. Interfejs znajdujący się w tej zakładce jest podzielony na trzy kolumny.



Rysunek 4.1: Zakładka 'Symulacja'

4.1.1 Zakładka 'Symulacja', Lewa Kolumna

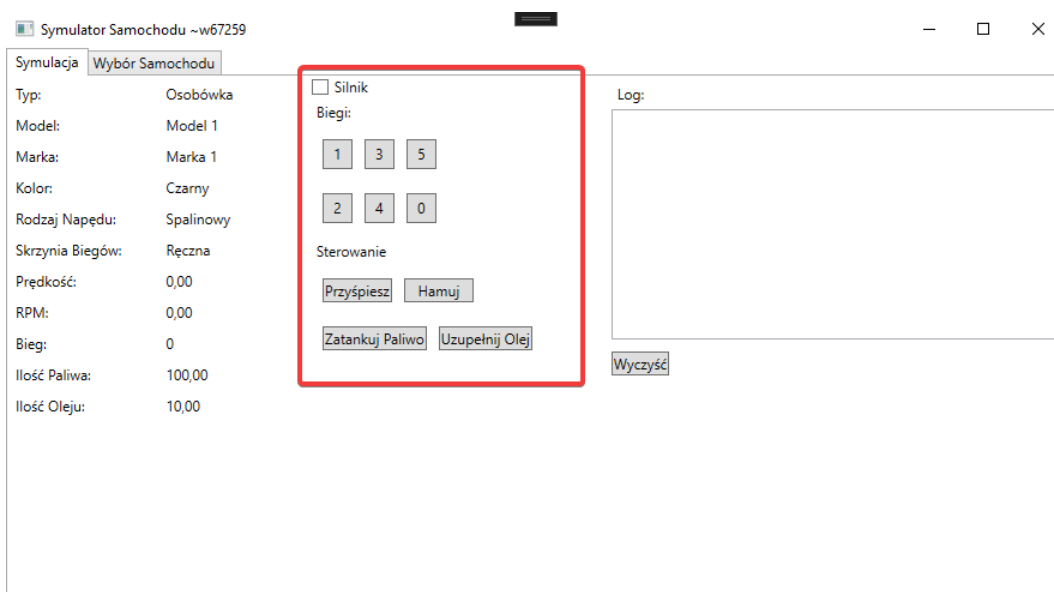
W kolumnie po lewej stronie możemy zobaczyć informacje o samochodzie (takie jak model, marka czy rodzaj napędu) oraz aktualne parametry samochodu (takie jak prędkość, ilość obrotów na minutę czy ilość paliwa lub oleju).



Rysunek 4.2: Zakładka 'Symulacja', Lewa Kolumna

4.1.2 Zakładka 'Symulacja', Środkowa Kolumna

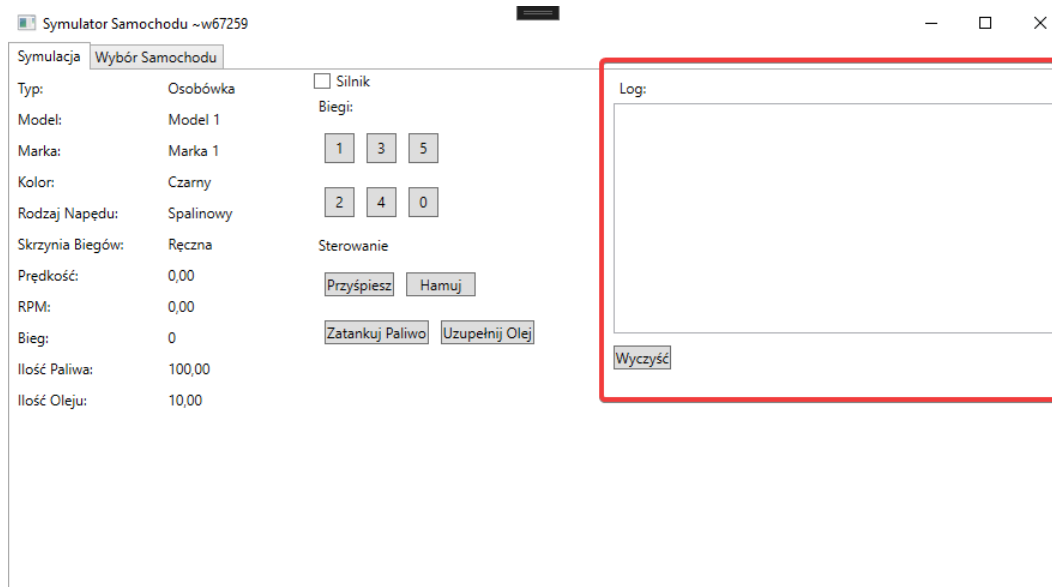
W środkowej kolumnie znajdują się elementy pozwalające na sterowanie symulacją. Takie jak przycisk do włączania i wyłączania silnika, zbiór przycisków pozwalających na zmianę biegu oraz przyciski, które odpowiadają za przyspieszanie, hamowanie oraz uzupełnianie paliwa i oleju.



Rysunek 4.3: Zakładka 'Symulacja', Środkowa Kolumna

4.1.3 Zakładka 'Symulacja', Prawa Kolumna

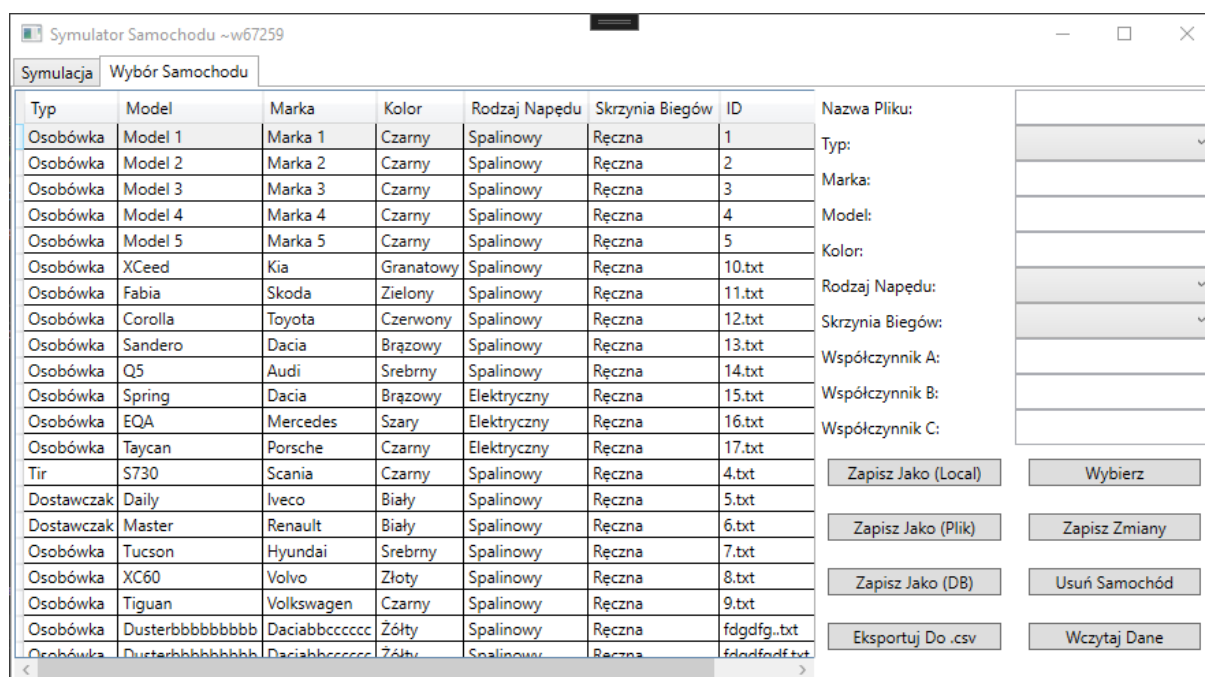
W kolumnie po prawej stronie znajduje się okienko z komunikatami, które mogą zostać wyświetlone w czasie działania programu. Znajduje się tam również przycisk, który pozwala na ręczne kasowanie wszystkich komunikatów. Użytkownik może dopisywać tak swoje uwagi.



Rysunek 4.4: Zakładka 'Symulacja', Prawa Kolumna

4.2 Zakładka 'Wybór Samochodu'

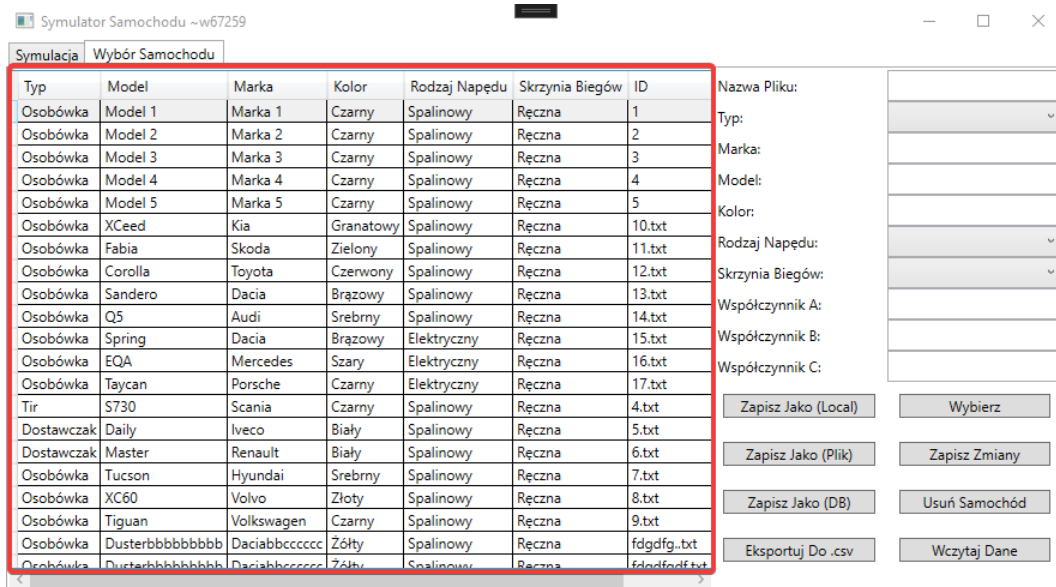
Zakładka "Wybór Samochodu" pozwala na wybór samochodu, który jest wykorzystywany w symulacji. Interfejs znajdujący się w tej zakładce jest podzielony na dwie kolumny.



Rysunek 4.5: Zakładka 'Wybór Samochodu'

4.2.1 Zakładka 'Wybór Samochodu', Lewa Kolumna

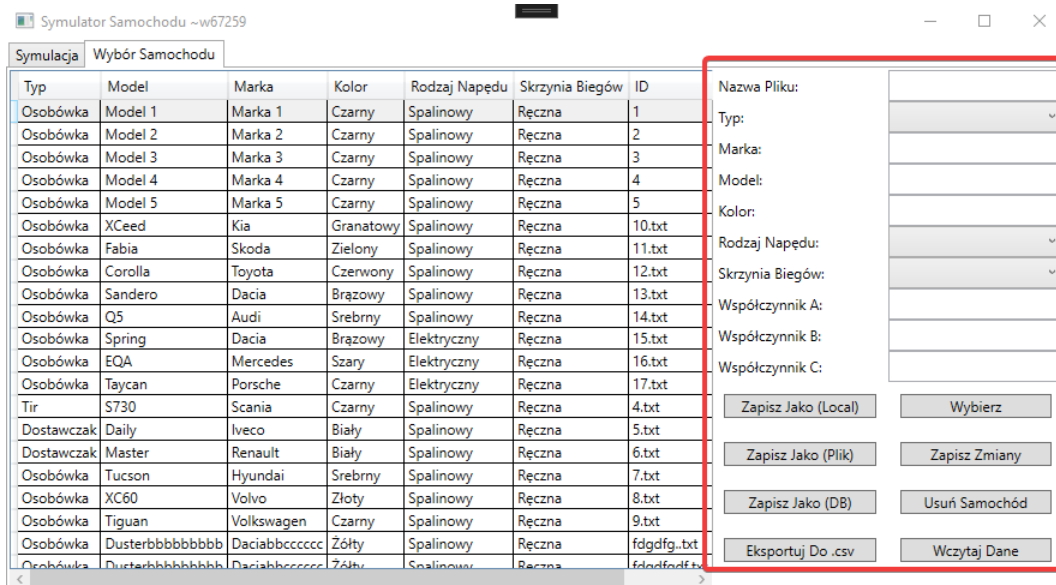
W kolumnie po lewej stronie znajduje się tabela z wszystkimi samochodami, które mogą zostać wykorzystane w symulacji. W tej tabeli znajdują się samochody ze wszystkich źródeł danych (tj. plików ".txt", pliku "Projekt_v1.db" oraz pliku "Dane.csv").



Rysunek 4.6: Zakładka 'Wybór Samochodu', Lewa Kolumna

4.2.2 Zakładka 'Wybór Samochodu', Prawa Kolumna

W kolumnie po prawej stronie znajdują się pola, w których wypisane są właściwości samochodu, który został zaznaczony w tabelce obok. Pod tymi wszystkimi polami znajdują się przyciski, które umożliwiają zapisywanie, modyfikowanie i usuwanie samochodów z symulacji.



Rysunek 4.7: Zakładka 'Wybór Samochodu', Prawa Kolumna

Rozdział 5

Podsumowanie

W ramach tego projektu przygotowałem program komputerowy, który pozwala na symulację samochodu, przyspieszanie, hamowanie, zmianę biegów oraz uzupełnianie paliwa i oleju. Program pozwala również na wybór samochodu z listy, która jest uzupełniona o pozycje z plików .txt, .csv oraz rekordy z bazy danych SQLite.

W przyszłości istnieje możliwość rozbudowy projektu o dodatkowe funkcje, takie jak dodatkowe rodzaje samochodów, dodatkowe rodzaje napędu lub skrzyni biegów, bardziej rozbudowany interfejs graficzny.

Bazę danych można by rozbudować o widoki czy kolejne tabele i utworzenie połączeń pomiędzy nimi, w celu normalizacji. Poprzez na przykład zmianę kolumny "Rodzaj_Napedu" na typ danych INT i przechowywanie w niej tylko odpowiednich ID oraz utworzenie powiązania do innej tabeli w której znajdowałyby się wyjaśnienia tych identyfikatorów.

Bibliografia

- [1] **Tutoriale z WPF**: <https://wpf-tutorial.com/> z dnia 25.01.2024
- [2] **Kursy C**: <https://www.w3schools.com/cs/index.php> z dnia 25.01.2024
- [3] **Strona SQLite**: <https://www.sqlite.org/index.html> z dnia 25.01.2024
- [4] **Tutoriale z SQLite**: <https://www.sqlitetutorial.net/> z dnia 25.01.2024
- [5] **Narzędzie SQLiteStudio**: <https://sqlitestudio.pl/> z dnia 25.01.2024
- [6] **Obsługa SQLite w VS**: <https://www.nuget.org/packages/Microsoft.Data.Sqlite/> z dnia 25.01.2024
- [7] **Obsługa SQLite w C**: <https://learn.microsoft.com/pl-pl/dotnet/standard/data/sqlite/?tabs=netcore-cli> z dnia 25.01.2024
- [8] **Wymagania sprzętowe Windows 10**: <https://support.microsoft.com/pl-pl/windows/wymagania-systemu-windows-10-6d4e9a79-66bf-7950-467c-795cf0386715> z dnia 25.01.2024
- [9] Jacek Matulewski, *C#: lekcje programowania: praktyczna nauka programowania dla platform .NET i .NET Core*, Helion, Gliwice 2021.
- [10] R. S. Miles, *C#: zacznij programować!*, Helion, Gliwice 2020.

Spis rysunków

2.1	Diagram Klas	6
3.1	Diagram Gantt	11
4.1	Zakładka 'Symulacja'	12
4.2	Zakładka 'Symulacja', Lewa Kolumna	13
4.3	Zakładka 'Symulacja', Środkowa Kolumna	13
4.4	Zakładka 'Symulacja', Prawa Kolumna	14
4.5	Zakładka 'Wybór Samochodu'	14
4.6	Zakładka 'Wybór Samochodu', Lewa Kolumna	15
4.7	Zakładka 'Wybór Samochodu', Prawa Kolumna	15