



# Samochód w mieście

Eryk Dobrosz

Wersja dokumentu: 2.0

07 lutego 2019

## Spis treści

1. Specyfikacja
  - 1.1. Opis biznesowy
  - 1.2. Wymagania funkcjonalne
  - 1.3. Wymagania niefunkcjonalne
2. Architektura rozwiązania i wykorzystane technologie
3. Klawiszologia

# 1. Specyfikacja

## 1.1. Opis biznesowy

Tworzona aplikacja ma na celu wyrenderowanie trójwymiarowej sceny przedstawiającej samochód w mieście. Aplikacja prezentuje różne modele cieniowania oraz pozwala użytkownikowi na modyfikowanie widocznej sceny przy pomocy transformacji macierzowych.

## 1.2. Wymagania funkcjonalne



Rysunek 1. Diagram przypadków użycia dla użytkownika

Tabela 1. Tabela opisów przypadków użycia dla użytkownika

Aktor	Nazwa	Opis
Użytkownik	Poruszanie samochodem	Użytkownik może jeździć samochodem po scenie przy pomocy klawiatury.
	Zmiana aktywnej kamery	Użytkownik może wybrać kamerę która ma być wykorzystana do renderowania sceny (nieruchoma, skierowana w stronę samochodu, poruszająca się wraz z samochodem) przy pomocy klawiatury.
	Zmiana trybu cieniowania	Użytkownik może wybrać który algorytm cieniowania będzie wykorzystywany do renderowania (cieniowanie Gourauda lub Phonga) przy pomocy klawiatury.
	Zmiana kierunku świecenia reflektorów	Użytkownik może zmienić kierunek świecenia reflektorów związanych z samochodem przy pomocy klawiatury.
	Zmiana pory dnia	Użytkownik może zmienić porę dnia (oświetlenie sceny) przy pomocy klawiatury.
	Zmiana odległości zanikania świateł	Użytkownik może wybrać dystans, który będzie odpowiadał za zanikanie świateł w zależności od odległości od aktywnej kamery.
	Zmiana intensywności mgły	Użytkownik może wybrać intensywność mgły, która będzie maskowała bardziej oddalone od kamery obiekty.
	Reset aplikacji	Użytkownik może przywrócić ustawienia początkowe aplikacji (wraz z pozycją samochodu) przy pomocy klawiatury.

### 1.3. Wymagania niefunkcjonalne

Obszar wymagań	Nr wymagania	Opis
Użyteczność (Usability)	1	Aplikacja działa w oknie w rozdzielczości natywnej ekranu, rozmiar i proporcje okna można zmienić bez zniekształceń obrazu.
Niezawodność (Reliability)	2	Aplikacja w przypadku błędu powinna go obsłużyć i poinformować o przyczynie jeśli dany błąd uniemożliwia dalszą pracę aplikacji.
Wydajność (Performance)	3	Aplikacja powinna płynnie renderować zadaną scenę i względnie szybko importować żądane modele.
Utrzymanie (Supportability)	4	Wraz z aplikacją zostanie dostarczona klawiszologia ze zwięzłymi objaśnieniami wypisanych akcji.

## 2. Architektura rozwiązania i wykorzystane technologie

Aplikacja została utworzona w języku C++ przy użyciu technologii OpenGL z wykorzystaniem bibliotek pomocniczych GLFW (do tworzenia okna oraz kontekstu OpenGL) oraz GLAD (do załadowania wskaźników do funkcji OpenGL'owych). Do rysowania obiektów zostały wykorzystane shadery (vertex shader oraz fragment shader) napisane w języku GLSL, w celu wykorzystania mocy obliczeniowej karty graficznej. Do operacji na macierzach i wektorach została wykorzystana biblioteka GLM (OpenGL Mathematics). Do importowania modeli 3D utworzonych za pomocą zewnętrznych aplikacji (np. Blender) została wykorzystana biblioteka Assimp (Open Asset Import Library).

Struktura sceny opiera się na klasie Node, która przechowuje informacje o transformacjach danego obiektu oraz powiązania z węzłami potomnymi oraz węzłem rodzica. Gdy dany Node jest renderowany, renderowane są też wszystkie jego dzieci. Za kamery odpowiada klasa Camera, która przechowuje pozycję oraz rotację kamery, oraz udostępniająca funkcje pomocnicze (np. `GetViewMatrix()`).

Wykorzystywane są też klasy i struktury pomocnicze, służące do podzielenia kodu bezpośrednio związanego z technologią OpenGL. Takimi klasami/strukturami są np. `PointLight`, `Mesh`, `Model`, przechowujące odpowiednie im informacje, a także zarządzające przypisywaniem zmiennych

uniformowych do shaderów, czy też przechowujące informacje o Vertex Array Objects dla odpowiednich zestawów wierzchołków.

Wykorzystany w programie model samochodu w formacie .obj został pobrany ze strony:  
<https://free3d.com/3d-model/chevrolet-camaro-ss-coupe-373476.html>

Pozostałe modele zostały przygotowane w programie Blender, a następnie wyeksportowane do plików .obj zawierających informacje o wierzchołkach oraz .mtl zawierających informacje o użytych w modelu materiałach.

### 3. Klawiszologia

- **1, 2, 3** - zmiana aktywnej kamery
- **W, S, A, D** - poruszanie pierwszą kamerą (tylko gdy aktywna jest kamera 1)
- **Ruch myszką** – obracanie pierwszą kamerą (tylko gdy aktywna jest kamera 1)
- **U, J, H, K** - poruszanie samochodem
- **4, 5** - zmiana intensywności mgły dla świateł (4 zwiększa intensywność mgły – zmniejsza zasięg widzenia)
- **Strzałki w górę i w dół** - zmiana intensywności mgły dla nieświecących obiektów (strzałka w dół zwiększa intensywność mgły – zmniejsza zasięg widzenia)
- **Strzałki w lewo i w prawo** - zmiana kierunku świecenia reflektorów samochodu
- **G, P** - zmiana modelu cieniowania (G – Gouraud, P – Phong)
- **N** - zmiana oświetlenia dzień/noc
- **R** - reset stanu aplikacji
- **ESC** – wyjście z aplikacji