

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

Grafika komputerowa 1  
Projekt 4: Grafika 3D

*Mustangiem przez Dzikie Zachód*  
Symulacja jazdy samochodem

Maja Kabus

Wersja 2.0  
04.02.2019 r.

# Spis treści

<b>1. Specyfikacja</b>	<b>2</b>
1.1 Opis biznesowy	2
1.2 Wymagania funkcjonalne	3
1.3 Wymagania niefunkcjonalne	4
<b>2. Architektura rozwiązania</b>	<b>5</b>
<b>3. Instrukcja obsługi</b>	<b>6</b>

# 1. Specyfikacja

## 1.1 Opis biznesowy

Aplikacja służy do możliwie realistycznej symulacji jazdy samochodem przez bezdroża Dzikiego Zachodu. Pojazd jest sterowany przez użytkownika.

## 1.2 Wymagania funkcjonalne

Poniższy diagram oraz tabela przedstawiają funkcje symulacji dostępne dla użytkownika.

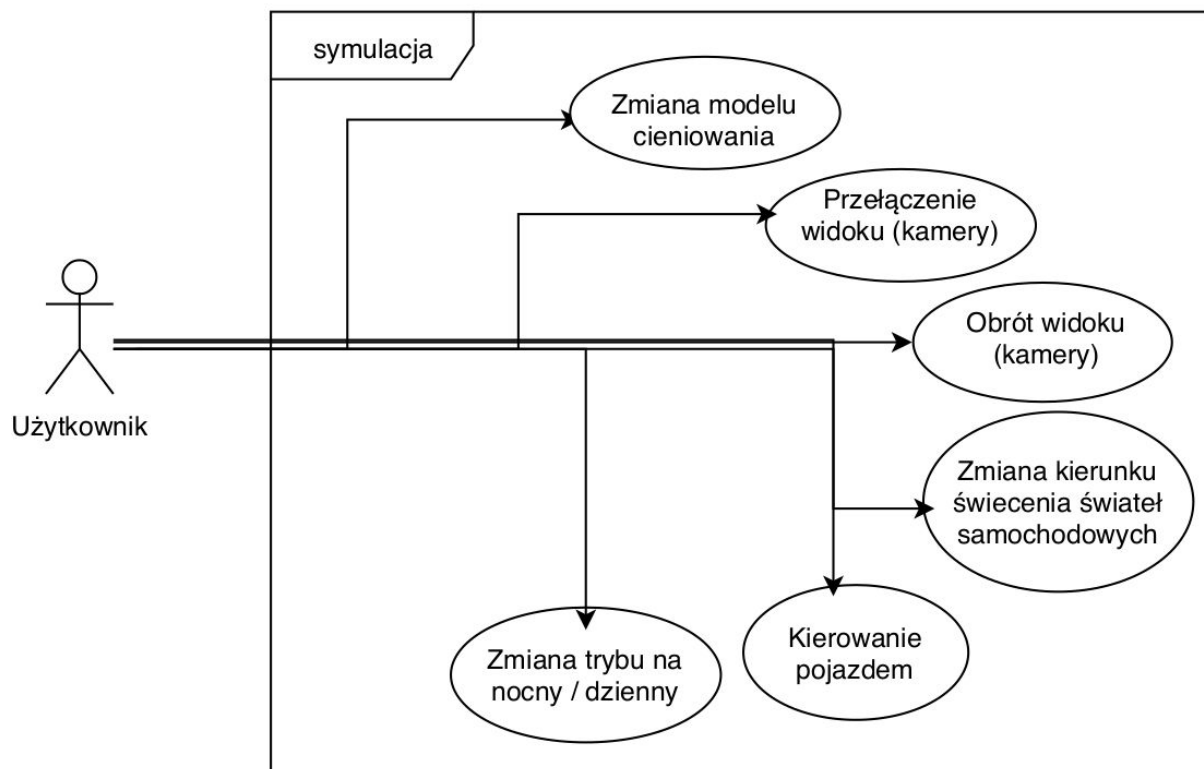


Diagram 1. Diagram przypadków użycia dla użytkownika aplikacji

Nazwa	Opis
Zmiana modelu cieniowania	Przełączanie między cieniowaniem Gorauda a cieniowaniem Phong.
Przełączanie widoku (kamery)	Przełączanie między 3 kamerami: z perspektywy ruszającego się pojazdu, nieruchoma kamera obserwująca scenę i nieruchoma kamera śledząca pojazd.
Obrót widoku (kamery)	Sterowanie kamerą za pomocą myszki.
Zmiana kierunku świecenia świateł samochodowych	Obracanie świateł samochodowych przy pomocy klawiatury.
Kierowanie pojazdem	Sterowanie samochodem za pomocą klawiatury.
Zmiana trybu na nocny / dzienny	Przełączanie oświetlenia na nocne i dzienne.

Tabela 1. Opisy przypadków użycia dla użytkownika aplikacji

### 1.3 Wymagania niefunkcjonalne

Poniższa tabela przedstawia wymagania niefunkcjonalne według modelu FURPS+.

Obszar wymagań	Numer wymagania	Opis
Funkcjonalność (Functionality)	1	Przedstawiona w poprzednim punkcie
Użyteczność (Usability)	2	Aplikacja otwiera się w pojedynczym oknie 800x600 px. Można rozmiar okna zwiększyć lub zmaksymalizować okno.
	3	Intuicyjne i łatwe sterowanie pojazdem oraz kamerami.
Niezawodność (Reliability)	4	Program zawczasu sprawdza dostępność wszystkich używanych modeli / tekstur i w przypadku braku kończy pracę przed właściwym rozpoczęciem symulacji.
	5	Symulacja nie kończy się bez uprzedniej zgody użytkownika.
	6	Możliwe błędy w działaniu są ukryte przed użytkownikiem i obsługiwane tak, by nie zakłócać działania symulacji.
Wydajność (Performance)	7	Wydajność zależy od urządzenia, ale powinna umożliwiać ciągły ruch samochodu.
	8	Symulacja bez zacinania zarządza min. 500 różnymi obiektami na scenie.
Utrzymanie (Supportability)	9	Instrukcja obsługi dołączona do aplikacji.

Tabela 2. Opisy przypadków użycia dla użytkownika aplikacji

## 2. Architektura rozwiązania

Aplikacja została stworzona w języku C++ z wykorzystaniem API OpenGL i pomocniczych bibliotek GLFW i GLM. Dodatkowo, shadery napisano w języku GLSL, a modele obiektów są przetwarzane w plikach .obj i .mtl i ładowane przy pomocy tinyobjloader<sup>1</sup>. Wszystkie potrzebne pliki biblioteczne są dołączone do paczki z programem.

Powyższe rozwiązania umożliwią przenośność oraz wydajne działanie aplikacji. Ponadto, warstwa obliczeniowa operująca na klasach C++ jest niezależna od przetwarzanych tekstur czy obiektów.

Modele na licencji Creative Commons pobrano z portalu SketchFab. Potrzebne pliki modeli i tekstur są ładowane za pomocą singletona Loader na początku działania programu, natomiast shaderami zarządza ShaderProgram i wyspecjalizowane klasy do poszczególnych shaderów. Modele są wczytane jako obiekty klasy Model, która zapamiętuje informacje wczytane z plików oraz klasy Entity, która zawiera informacje związane z rolą obiektu w aplikacji. Po Entity dziedziczy klasa Player obsługująca poruszający się samochód.

Światła są zapisane w prostej strukturze Light zawierającej potrzebne parametry. W programie zastosowano kierunkowe źródło światła do reprezentacji słońca bądź światła nocnego oraz reflektor jako światła pojazdu. Światła samochodowe mogą być sterowane klawiaturą, istnieje również opcja włączenia światel policyjnych - niebieskiego, obracającego się reflektora.

Świat w aplikacji można obserwować za pomocą 3 kamer:

- tracking - z perspektywy kierowcy pojazdu
- moving - kamera zawieszona z tyłu pojazdu, śledząca samochód, może poruszać się dookoła pojazdu
- standing - nieruchoma kamera obserwująca scenę z wysoka

Kamery są zarządzane klasą Camera z wyspecjalizowanymi podklasami dla każdego typu kamery. We wszystkich trybach możliwy jest obrót kamery, w moving i standing również zoom.

---

<sup>1</sup> <https://github.com/syoyo/tinyobjloader>

### 3. Instrukcja obsługi

Tylko kamery są obsługiwane myszką. Kamerę moving można obrócić poprzez ruch myszką przy wciśniętym lewym przycisku, pozostałe kamery nie wymagają naciskania przycisku. Zoom uzyskuje się przewijaniem myszy.

Pozostałe funkcje obsługiwane są klawiaturą:

- strzałki ←↑→↓ lub klawisze A, W, D, S - sterowanie samochodem
  - w kodzie dostępne są dwie metody sterowania: naiwna oraz fizyczna
  - domyślnie, w metodzie naiwnej, pojazd porusza się ze stałą prędkością
  - przy włączonej (w kodzie) fizycznej metodzie, strzałki ↑↓ (klawisze W, S) przyspieszają i zwalniają pojazd, nie jest możliwe skręcanie przy zerowej prędkości
- J, I, L, K - sterowanie światłami samochodowymi
  - I / K - obrót góra / dół
  - J / L - obrót lewo / prawo
- P - przełączanie między cieniowaniem Phong'a a cieniowaniem Gouraud
- F - włączanie i wyłączanie efektu mgły
- Z - włączanie trybu dziennego
- C - włączanie trybu nocnego
- X - wyłączanie skyboxa bez zmiany trybu
  - ponowne włączenie możliwe poprzez przejście do trybu dziennego / nocnego
- Q - włączanie i wyłączanie świateł policyjnych
- T - kamera tracing
- Y - kamera moving
- U - kamera standing