Spis treści

[0 Abstrakt 1](#_Toc502881194)

[1 Wstęp 2](#_Toc502881195)

[1.1 Charakterystyka problematyki 2](#_Toc502881196)

[1.2 Uzasadnienie wyboru tematu pracy 2](#_Toc502881197)

[1.3 Cel i zakres pracy 2](#_Toc502881198)

[2 Przegląd stosowanych technologii 3](#_Toc502881199)

[2.1 OpenGL 3](#_Toc502881200)

[2.1.1 Czym jest OpenGL 3](#_Toc502881201)

[2.1.2 Wersje OpenGL 3](#_Toc502881202)

[2.1.3 OpenGL jako maszyna stanów 3](#_Toc502881203)

[2.1.4 Kolejność renderowania w OpenGL 3](#_Toc502881204)

[2.1.5 GLU 3](#_Toc502881205)

[2.1.6 Zarządzanie stanem 4](#_Toc502881206)

[2.1.7 Transformacja modelująca 4](#_Toc502881207)

[2.1.8 Transformacja obserwatora 4](#_Toc502881208)

[2.1.9 Transformacja. Rzutowanie 4](#_Toc502881209)

[2.1.10 Oświetlenie 4](#_Toc502881210)

[2.1.11 Właściwości powierzchniowe. Materiały. 4](#_Toc502881211)

[2.1.12 Teksturowanie 4](#_Toc502881212)

[2.2 Wspierane formaty 4](#_Toc502881213)

[2.2.1 Omówienie wykorzystanych formatów plików 4](#_Toc502881214)

[2.2.2 Uzasadnienie wyboru 4](#_Toc502881215)

[2.2.3 Definicja geometrii sceny – format OBJ 4](#_Toc502881216)

[2.2.4 Definicja materiałów – format MTL 4](#_Toc502881217)

[2.2.5 Tekstury – popularne formaty rastrowe 4](#_Toc502881218)

[3 Architektura i Implementacja rozwiązania 5](#_Toc502881219)

[3.1 Analiza wymagań 5](#_Toc502881220)

[3.2 Architektura rozwiązania 5](#_Toc502881221)

[3.3 Opis komponentów, uml 5](#_Toc502881222)

[3.4 Wykorzystanie technologie 5](#_Toc502881223)

[3.5 Problemy napotkane podczas implementacji 5](#_Toc502881224)

[4 Instrukcja 5](#_Toc502881225)

[4.1 Instalacja 5](#_Toc502881226)

[4.2 Konfiguracja 5](#_Toc502881227)

[4.3 Instrukcja użytkowania 5](#_Toc502881228)

[5 Podsumowanie i wnioski 5](#_Toc502881229)

[5.1 Czego się nauczyłem 5](#_Toc502881230)

[5.2 Dalszy rozwój 5](#_Toc502881231)

[5.2.1 Wykorzystanie biblioteki Prism 5](#_Toc502881232)

[5.2.2 Rozbudowa interfejsu użytkownika 5](#_Toc502881233)

[5.2.3 Pokrycie kodu testami 5](#_Toc502881234)

[6 Bibliografia 6](#_Toc502881235)

# Abstrakt

|  |  |
| --- | --- |
| Celem pracy jest stworzenie narzędzia dydaktycznego mającego wspomagać nauczanie grafiki komputerowej na Politechnice Wrocławskiej. Jej zakres obejmuje projekt i implementację programu pozwalającego w prosty sposób zacząć przygodę z grafiką komputerową. Ma on prezentować podstawowe funkcjonalności bibliotek OpenGL. Powinien pozwalać  na wczytanie sceny i eksperymentowanie  z jej parametrami z poziomu graficznego interfejsu użytkownika. Program musi także oferować możliwość wygenerowania kodu który po skompilowaniu utworzy scenę odpowiadającą tej skonfigurowanej przez użytkownika. Do programu dołączona zostanie instrukcja oraz krótki opis funkcjonalności bibliotek OpenGL prezentowanych przez program. Zostaną one zawarte w tej pracy. | The aim of this thesis is creation of a didactic tool facilitating teaching process of computer graphics on Wroclaw University of Science. Its scope consists of design and implementation of an application allowing to easily start an adventure with computer graphics. It’s expected to present basic features of an OpenGL libraries. It should allow reading a scene definition and playing with its parameters by a graphic user interface. Moreover, program must allow generating code compilation of which generates a scene corresponding to the one configured by the user. A manual and a brief description of OpenGL libraries features presented by the program will be included in this thesis. |

# Wstęp

## Charakterystyka problematyki

Stare chińskie przysłowie głosi:

„Usłyszałem i zapomniałem. Zobaczyłem i zapamiętałem. Zrobiłem i zrozumiałem.”

Według tej zasady, najlepszą metodą przyswajania wiedzy jest nauka interaktywna.   
Dotychczas dostępne metody nauki grafiki komputerowej obejmują:

* prezentacje multimedialne dostarczane przez prowadzącego,
* literaturę tematyczną,
* wideo-poradniki dostępne w internecie,
* tutoriale dostępne w internecie,
* laboratoria powiązane z kursem.

Pierwsze trzy wymienione opcje skupiają się na biernym odbiorze i nie są przystosowane   
do efektywnej pracy z kodem oraz prezentacji szczegółów implementacyjnych omawianych tematów. Kolejne dwa podejścia wymagają posiadania odpowiednio skonfigurowanego środowiska programistycznego (IDE, kompilatory, biblioteki OpenGL).

Jak widać każda z wyżej wymienionych metod posiada braki.

## Uzasadnienie wyboru tematu pracy

Temat pracy dyplomowej wzbudził moje zainteresowanie ze względu na konieczność dogłębnego zapoznania się z biblioteką OpenGL podczas jego realizacji. Zaintrygowała mnie również możliwość zbudowania narzędzia które wspomagać będzie proces przyswajania wiedzy przez kolejne pokolenia.

## Cel i zakres pracy

Celem pracy jest stworzenie narzędzia wspomagającego proces nauczania grafiki komputerowej na Politechnice Wrocławskiej. Jego osiągnięcie ma zapewnić projekt, implementacja i dokumentacja programu komputerowego prezentującego w interaktywny sposób funkcjonalności oferowane przez biblioteki OpenGL.

Zadaniem programu będącego jednym z artefaktów tej pracy jest umożliwienie interaktywnej nauki bez konieczności skomplikowanej konfiguracji środowiska. Ma on pozwalać na eksperymentowanie z podstawowymi funkcjonalnościami bibliotek OpenGL takimi jak:

* modyfikacja oferowanych przez OpenGL zmiennych stanu
* transformacja projekcji
* transformacje modelująca i obserwatora
* teksturowanie
* cieniowanie / własności powierzchniowe

Dokumentacja powinna zawierać:

* instrukcję instalacji,
* instrukcję użytkowania,
* krótki opis wspieranych przez program funkcjonalności OpenGL.

# Przegląd stosowanych technologii

## OpenGL

### Czym jest OpenGL

Przed przystąpieniem do jego omawiania warto wyjaśnić czym jest OpenGL.   
Open Graphics Library jest to interfejs programowania aplikacji (API) służący do renderowania grafiki. Składa się on z około 250 funkcji pozwalających na tworzenie grafiki 2D oraz 3D. Oferuje on uzyskanie przyspieszenia sprzętowego poprzez wykorzystanie procesora graficznego (GPU).

Ponieważ OpenGL jest interfejsem istnieje wiele jego implementacji dla różnych języków programowania. Mogą się one różnić wydajnością jednak ich sposób ich działania pozostaje taki sam.

### Wersje OpenGL

W celu czytelnego zaprezentowania podstaw grafiki komputerowej wykorzystana została stara wersja OpenGL ( Fixed Function Pipeline / Legacy OpenGL). Jej używanie jest niezalecane ze względu na ograniczone możliwości w porównaniu do nowszych wersji API.

Taka wersja świetnie się jednak nadaje by przedstawić podstawowy potok renderowania (Rendering Pipeline) bez dodatkowego narzutu wprowadzanego przez shader’y. Program ten nie prezentuje jak należy tworzyć aplikacje korzystając z OpenGL. Powinien być stosowany tylko w celu zrozumienia mechanizmów występujących w grafice komputerowej które w wielu wypadkach zostały już zastąpione przez bardziej elastyczne   
i wydajne rozwiązania. Niemniej poznanie podstawowych idei omówionych w programie zapewnia solidne podstawy z zakresu grafiki komputerowej które pozwolą bez problemu pojąć koncepty i założenia stosowane w nowszych wersjach OpenGL.

############## ?? VULCAN ?? #############

### OpenGL jako maszyna stanów

### Kolejność renderowania w OpenGL

123

### GLU

Ponieważ OpenGL zawiera jedynie zestaw podstawowych funkcji istnieje wiele bibliotek które wspomagają jego wykorzystanie. Jedną z nich jest OpenGL Utility   
Library (GLU). Zawiera ona zestaw około 50 funkcji ułatwiających korzystanie z OpenGL i jest standardową częścią każdej jego implementacji.

### Zarządzanie stanem

### Transformacja modelująca

### Transformacja obserwatora

### Transformacja. Rzutowanie

### Oświetlenie

### Właściwości powierzchniowe. Materiały.

### Teksturowanie

## Wspierane formaty

### Omówienie wykorzystanych formatów plików

### Uzasadnienie wyboru

### Definicja geometrii sceny – format OBJ

### Definicja materiałów – format MTL

### Tekstury – popularne formaty rastrowe

# Architektura i Implementacja rozwiązania

## Analiza wymagań

## Architektura rozwiązania

## Opis komponentów, uml

## Wykorzystanie technologie

## Problemy napotkane podczas implementacji

# Instrukcja

## Instalacja

## Konfiguracja

## Instrukcja użytkowania

+ screeny

# Podsumowanie i wnioski

## Czego się nauczyłem

## Dalszy rozwój

### Wykorzystanie biblioteki Prism

### Rozbudowa interfejsu użytkownika

### Pokrycie kodu testami

# Bibliografia

~15 pozycji