

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA
SPECJALNOŚĆ: GRAFIKA I SYSTEMY MULTIMEDIALNE

PRACA DYPLOMOWA
INŻYNIERSKA

Interactive 3D visualization of geographical data
based on open sources using web technologies.

Interaktywna wizualizacja 3D danych
geograficznych z otwartych źródeł
z wykorzystaniem technologii webowych.

AUTOR:

Damian Koper

PROWADZĄCY PRACĘ:

Dr inż. Marek Woda

OCENA PRACY:

Spis treści

1. Wymagania	6
1.1. Twórca wizualizacji	6
1.1.1. Wymagania funkcjonalne	6
1.1.2. Wymagania niefunkcjonalne	7
1.2. Odbiorca wizualizacji	7
1.2.1. Wymagania funkcjonalne	7
1.2.2. Wymagania niefunkcjonalne	7
1.3. Aplikacja	7
1.3.1. Wymagania niefunkcjonalne	7
2. Architektura aplikacji	8
Literatura	9

Spis rysunków

2.1. Schemat ogólny komponentów aplikacji.	8
----------------------------------------------------	---

Spis tabel

Spis listingów

Rozdział 1

Wymagania

Ze względu na możliwy podział funkcjonalności projektu na wiele typów, zdefiniowano następujące pojęcia:

1. Silnik - zbiór komponentów odpowiedzialnych za definicję i wyświetlenie wizualizacji.
2. Wizualizacja - konfigurowalny widok przedstawiający obiekty, których położenie zdefiniowano za pomocą współrzędnych geograficznych, na powierzchni sfery.
3. Aplikacja - uruchomiona w przeglądarce użytkownika strona umożliwiająca wybór i wyświetlenie wizualizacji.

Silnik dostarcza komponenty i interfejs programistyczny, dzięki którym można definiować, wyświetlać i zarządzać wizualizacją. Pozwala także na zdefiniowanie wielu niezależnych wizualizacji. Z tego powodu można wyróżnić dwa typy użytkowników:

1. Twórcę wizualizacji,
2. Odbiorcę wizualizacji.

Wymagania aplikacji zostały zdefiniowane z podziałem na typ użytkownika. Struktura danych definiująca renderowany obraz, zwana dalej będzie sceną.

1.1. Twórca wizualizacji

1.1.1. Wymagania funkcjonalne

1. Twórca może zdefiniować metadane wizualizacji określone przez interfejs Silnika.
2. Twórca może zdefiniować statyczną scenę określając położenie obiektów na sferze z wykorzystaniem długości i szerokości geograficznej oraz wysokości nad poziomem morza.
3. Twórca do definicji sceny może wykorzystać interfejs tworzenia obiektów dostarczony przez aplikację lub załadować obiekty, materiały i tekstury z zewnętrznego źródła.
4. Twórca może zagnieźdzać sceny predefiniowane w silniku, oraz sceny wcześniej stworzonych przez siebie.
5. Twórca może parametryzować sceny w celu określonej ich modyfikacji w procesie zagnieźdżania.
6. Twórca może określić parametry początkowe obserwatora, dynamikę i zakres jego ruchów:
 1. położenie,
 2. prędkość i przyspieszenie ruchu,
 3. ograniczeni przybliżenia,
 4. ograniczeni pozycji.

7. Twórca może zdefiniować wygląd i funkcjonalność panelu kontrolnego. Panel ten służyć będzie do zmiany parametrów wizualizacji i obsługiwany będzie przez odbiorcę.
8. Twórca, poprzez interfejs programistyczny dostarczony przez silnik, może aktualizować scenę w dowolnym momencie, określonym przez siebie w definicji wizualizacji.
9. Twórca może definiować zachowania, które będą odpowiedzią na zdarzenia związane z poruszaniem się po scenie generowane przez odbiorcę.

1.1.2. Wymagania niefunkcjonalne

1. Silnik powinna definiować i w sposób jasny przekazywać potencjalnemu twórcy akceptowalną strukturę danych, plików i katalogów, określającą jedną wizualizację.
2. Włączenie zdefiniowanej wizualizacji do ich zbioru w aplikacji powinno ustanowione być tylko w jednym miejscu poprzez prosty interfejs.
3. Dane wizualizacji muszą być ładowane asynchronicznie. Dane źródłowe definiujące scenę mogą być przetwarzane po stronie odbiorcy lub być przetworzone wcześniej i pobrane.

1.2. Odbiorca wizualizacji

1.2.1. Wymagania funkcjonalne

1. Odbiorca może zobaczyć dane dostępnych wizualizacji.
2. Odbiorca może wyświetlić wybraną wizualizację.
3. Odbiorca może poruszać się po wizualizacji, zmieniając położenia kamery, używając myszki lub klawiatury.
4. Odbiorca może wyświetlić lub ukryć przyciski sterowania kamerą.
5. Odbiorca może wyświetlić lub ukryć panel sterujący wizualizacją dostarczony przez twórcę.
6. Odbiorca może ustawić poziom szczegółowości grafiki. Ustawienia te przekazywane są twórcy i mogą, ale nie muszą, być respektowane.

1.2.2. Wymagania niefunkcjonalne

1. Każda akcja użytkownika może zostać wykonana używając myszki lub równolegle klawiatury.

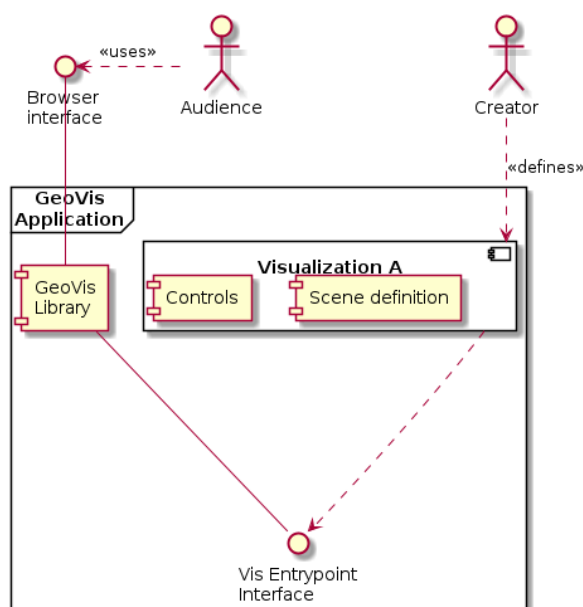
1.3. Aplikacja

1.3.1. Wymagania niefunkcjonalne

1. Aplikacja powinna być stroną typu Single Page Application.
2. Jeśli to możliwe aplikacja powinna wykorzystywać sprzętową akcelerację obliczeń graficznych.
3. Aplikacja powinna ustawiać i obsługiwać adres URL w przeglądarce jednoznacznie definiujący jej widok.

Rozdział 2

Architektura aplikacji



Rys. 2.1: Schemat ogólny komponentów aplikacji.

Literatura