

Proceduralne generowanie planet w czasie rzeczywistym  
w dynamicznie zmieniającej się skali

# Dokumentacja projektowa

---

MATEUSZ CHECHLIŃSKI

*Wersja 4.0*

20 stycznia 2015

Modyfikacje dokumentu			
Wersja	Data	Autor	Opis
1.0	22 listopada 2014	Mateusz Chechliński	Iteracja nr 1.
2.0	10 grudnia 2014	Mateusz Chechliński	Iteracja nr 2.
3.0	5 stycznia 2014	Mateusz Chechliński	Iteracja nr 3.
4.0	20 stycznia 2014	Mateusz Chechliński	Wersja ujednolicona z powodu braku zmian między iteracjami

# Spis treści

Spis treści	3
1. Streszczenie	4
2. Słownik pojęć	4
3. Wstęp	4
4. Architektura	5
5. Cykl życia	6
6. Interface	6
7. Technologie	6

# 1. Streszczenie

Niniejszy dokument stanowi projekt aplikacji do proceduralnego generowania planet w czasie rzeczywistym w dynamicznie zmieniającej się skali.

Aplikacja będzie działać w oparciu o prostą architekturę, w której biblioteka jQuery obsługuje komunikację z minimalistycznym serwerem HTTP, zaś biblioteka Three.js w oparciu o WebGL zapewni płynne wyświetlanie grafiki 3D.

Aplikacja będzie mieć bardzo minimalistyczny interfejs - całą przestrzeń zajmie wyrenderowany obraz, a sterowanie odbywać się będzie za pomocą myszy i klawiatury.

## 2. Słownik pojęć

**CPU** - *Central Processing Unit*, mikroprocesor

**GPU** - *Graphics Processing Unit*, procesor graficzny (potocznie: karta graficzna)

**Shader** - krótki program komputerowy, który oblicza właściwości pikseli oraz wierzchołków.

**Uniform** - globalna zmienna GLSL zadeklarowana ze słowem kluczowym „uniform”. Wykorzystywana w roli atrybutu przekazywanego do shadera.

**GLSL** - *OpenGL Shading Language*, język programowania potoku graficznego składniowo zbliżony do języka C, wykorzystywany przez bibliotekę OpenGL oraz WebGL.

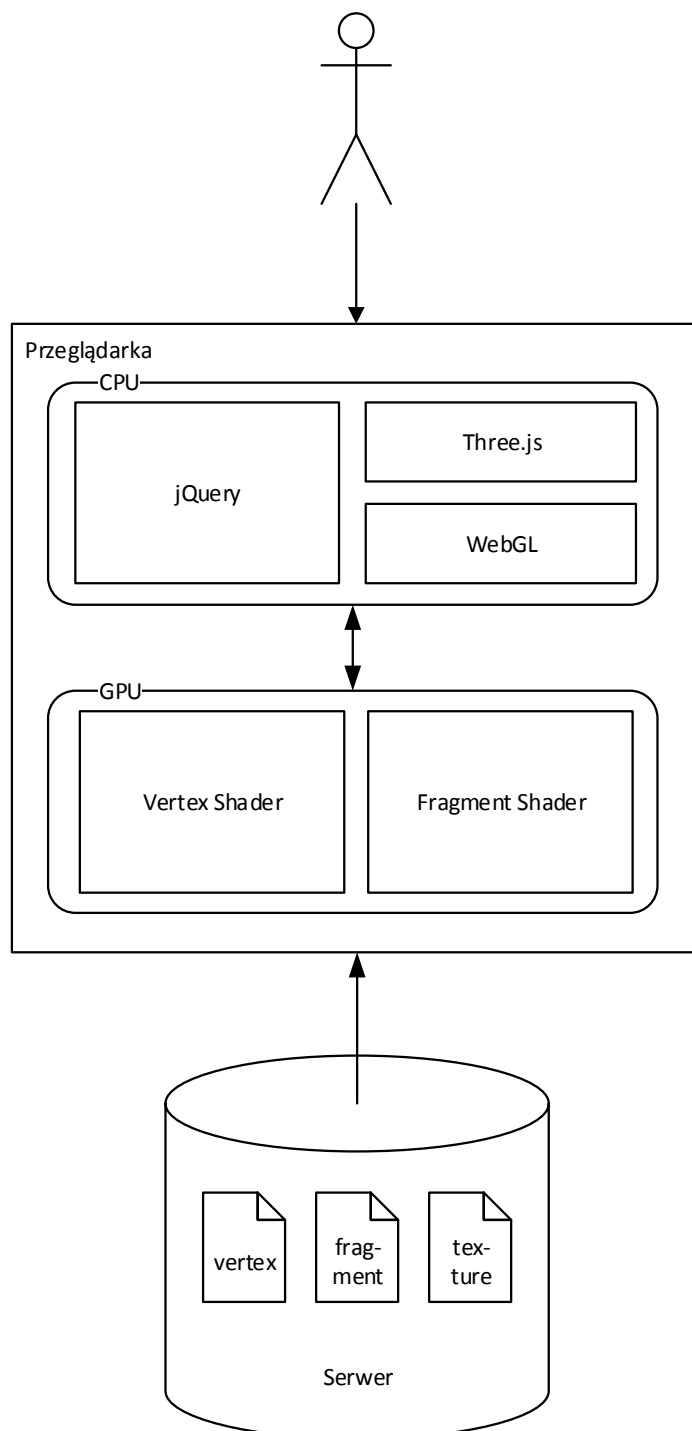
**AJAX** - *Asynchronous JavaScript and XML*, asynchroniczny JavaScript i XML. Technika tworzenia aplikacji internetowych, w której interakcja użytkownika z serwerem odbywa się bez przeładowywania całego dokumentu, w sposób asynchroniczny.

**Renderowanie** - (od ang. *rendering*), analiza modelu danej sceny oraz utworzenie na jej podstawie dwuwymiarowego obrazu wyjściowego w formie statycznej lub animacji.

## 3. Wstęp

Niniejszy dokument stanowi dokumentację projektową aplikacji do proceduralnego generowania planet w czasie rzeczywistym w dynamicznie zmieniającej się skali.

## 4. Architektura



Rys. 1 - Schemat architektury aplikacji

Aplikacja będzie działać w całości w przeglądarce internetowej. Dzięki wykorzystaniu biblioteki WebGL, możliwe będzie wykorzystanie mocy obliczeniowej GPU.

Wykorzystamy prosty serwer HTTP do serwowania plików, nie tylko przy uruchomieniu aplikacji, ale także w trakcie jej działania.

Za komunikację z serwerem odpowiadać będzie biblioteka jQuery, za pomocą której będziemy wysyłać zapytania AJAX do serwera.

Za grafikę trójwymiarową natomiast odpowiadać będzie biblioteka Three.js, działająca w oparciu WebGL.

GPU będzie wykonywał dwa shadery - Vertex Shader i Fragment Shader. Pierwszy z nich będzie odpowiedzialny za wyznaczenie położenia w przestrzeni dla każdego wierzchołka, a także za określenie wartości atrybutów do interpolacji we Fragment Shaderze. Fragment Shader posłuży do określania koloru każdego z pikseli.

## 5. Cykl życia

Ze względu na charakter aplikacji, jej cykl życia będzie bardzo minimalistyczny i będzie składał się z następujących etapów:

1. Inicjalizacja
2. Renderowanie (w pętli)

W pierwszej fazie aplikacja załaduje niezbędny pliki (shadery, tekstury itp.), zainicjuje scenę oraz wygeneruje planetę/gwiazdę na podstawie ziarna.

W drugiej fazie model planety/gwiazdy będzie renderowany przy pomocy GPU i wyświetlany użytkownikowi. Przerwanie pętli renderującej następuje wyłącznie przy zakończeniu pracy aplikacji.

## 6. Interface

Aplikacja będzie posiadać bardzo minimalistyczny interface. Całą dostępną powierzchnię strony zajmie kontrolka, na której będzie prezentowany wyrenderowany obraz. Sterowanie będzie się odbywać za pomocą myszy i klawiatury.

## 7. Technologie

W projekcie zostaną wykorzystane następujące technologie:

1. **Three.js** - jest lekka biblioteka JavaScript, wspierająca wiele przeglądarek i działająca w oparciu o WebGL. Pozwala na tworzenie trójwymiarowych animacji dzięki wykorzystaniu w przeglądarce mocy obliczeniowej karty graficznej.
2. **qUnit** - framework służący do testów jednostkowych dla aplikacji napisanych w JavaScript
3. **SimpleHttpServer** - prosty serwer Pythonowy, zostanie wykorzystany wyłącznie do serwowania plików do aplikacji w czasie jej działania.