

Proceduralne generowanie planet w czasie rzeczywistym
w dynamicznie zmieniającej się skali

Analiza wymagań

MATEUSZ CHECHLIŃSKI

Wersja 4.0

20 stycznia 2015

Modyfikacje dokumentu

Wersja	Data	Autor	Opis
1.0	15 listopada 2014	Mateusz Chechliński	Iteracja nr 1. - Planeta ziemio-podobna
1.1	20 listopada 2014	Mateusz Chechliński	Iteracja nr 1. - Dodano różne rodzaje planet i gwiazdy
2.0	8 grudnia 2014	Mateusz Chechliński	Iteracja nr 2. - Poziom szczegółów
3.0	3 stycznia 2014	Mateusz Chechliński	Iteracja nr 3. - Wiele planet
4.0	20 stycznia 2014	Mateusz Chechliński	Iteracja nr 4. - Oświetlenie

Spis treści

Spis treści	3
1. Streszczenie	4
2. Słownik pojęć	4
3. Wstęp	4
4. Iteracja nr 1	5
4.1 Cel iteracji	5
4.2 User stories	6
4.3 Analiza funkcjonalna	7
4.4 Podział prac	7
5. Iteracja nr 2	9
5.1 Cel iteracji	9
5.2 User stories	9
5.3 Analiza funkcjonalna	10
5.4 Podział prac	10
6. Iteracja nr 3	11
6.1 Cel iteracji	11
6.2 User stories	11
6.3 Analiza funkcjonalna	12
6.4 Podział prac	13
7. Iteracja nr 4	13
7.1 Cel iteracji	13
7.2 Analiza funkcjonalna	14
7.3 Podział prac	14

1. Streszczenie

Celem projektu jest stworzenie aplikacji webowej służącej do dynamicznego generowania planet, a następnie prezentowania ich w formie grafiki trójwymiarowej. Projekt podzielony jest na cztery iteracje, z czego każda stanowi zamkniętą część.

Iteracja 1. - Celem tego etapu jest opracowanie aplikacji generującej i wyświetlającej pojedynczą, trójwymiarową planetę lub gwiazdę. Powierzchnia planety/gwiazdy pokolorowane i ukształtowane jest w sposób odpowiadający jej rodzajowi i możliwie różnorodny. Użytkownik może ją obserwować z każdej strony za pomocą kamery swobodnie przemieszczającej się po orbicie.

Iteracja 2. - Celem tego etapu jest wzbogacenie aplikacji o algorytm Level of Details. Pozwoli to prezentować planety i gwiazdy nie tylko z każdej strony, ale też z każdej odległości od powierzchni.

Iteracja 3. - Celem tego etapu jest rozszerzenie funkcjonalności na wiele planet i gwiazd. Aplikacja wygeneruje kosmos składający się z co najmniej kilkudziesięciu obiektów. Użytkownik będzie mógł swobodnie po niej nawigować, a poziom detali będzie dynamicznie dopasowywany do położenia kamery w celu zachowania jak największej liczby szczegółów przy minimum 30 klatkach na sekundę.

Iteracja 4. - Celem tego etapu jest rozszerzenie funkcjonalności o efekty graficzne związane z oświetleniem planet przez gwiazdy.

Analiza wymagań kolejnych iteracji będzie dodawana sukcesywnie w miarę postępów prac.

2. Słownik pojęć

Ziarno (Seed) - pewna liczba, na podstawie której generowana będzie planeta lub gwiazda. Dla danego ziarna wygenerowany obiekt zawsze będzie dokładnie taki sam.

Level of Details - (z ang.: *poziom szczegółów*) pojęcie w grafice komputerowej, określające poziom skomplikowania trójwymiarowego obiektu znajdującego się w odpowiedniej odległości od punktu widzenia. Technika LOD zwiększa wydajność renderingu poprzez spadek ilości wyświetlanych punktów (źródło: http://pl.wikipedia.org/wiki/Level_of_detail)

FPS - (ang.: *Frames per second*) Liczba klatek wyświetlanych w ciągu sekundy. Jest ona miarą płynności wyświetlania ruchomych obrazów.

3. Wstęp

Niniejszy dokument stanowi pełną analizę wymagań projektu. W związku z wybraną metodyką prac, polegającą na podzieleniu zadania na cztery osobne iteracje, treść dokumentu będzie stopniowo uzupełniana o rozdziały odpowiadające kolejnym iteracjom.

4. Iteracja nr 1

4.1 Cel iteracji

Celem pierwszej iteracji jest opracowanie aplikacji webowej dynamicznie generującej pojedynczą planetę lub gwiazdę. Aplikacja na tym etapie ma mieć następujące cechy:

1. Planeta lub gwiazda jest generowana na podstawie ziarna (seed). Ziarno jest losowane za każdym razem, gdy przeładowywana jest strona.
2. Planeta może należeć do jednego z poniższych rodzajów
 - a. ziemio-podobna
 - b. pustynna
 - c. lodowa
 - d. lawowa
3. Powierzchnia planety ziemio-podobnej ma mieć następujące cechy:
 - a. Na planecie występują zbiorniki wodne (umownie: „morza” i „oceany”).
 - b. Można wyróżnić duże powierzchnie stałego lądu (umownie: „kontynenty”) jak również maleńkie „wysepki”.
 - c. Powierzchnia lądu jest zróżnicowana jeśli chodzi o wysokość nad poziomem wód.
 - d. Woda oznaczona jest kolorem niebieskim, lądy kolorowane są na podstawie wysokości nad poziomem morza: kolor zielony dla nizin, kolor ciemnobrazowy dla gór - analogicznie do tego, jak zaznacza się powierzchnię na mapach topologicznych.
4. Planeta lawowa poprzecinana jest siecią magmowych rzek.
5. Planeta lodowa charakteryzuje się nierówną powierzchnią i różną grubością powierzchni lodu, co przekłada się na jej kolor.
6. Planeta pustynna ma płaską powierzchnię, ponad którą wznoszą się góry, oznaczone ciemniejszym kolorem.
7. Gwiazda ma płaską powierzchnię, ale jest animowana.
8. To czy zostanie wygenerowana gwiazda, czy planeta oraz ewentualnie jakiego rodzaju jest planeta, jest losowe.
9. Planeta lub gwiazda znajduje się w pustej przestrzeni i jest oświetlona równomiernie z każdej strony.
10. Użytkownik może poruszać się wyłącznie po orbicie planety lub gwiazdy. Nie ma możliwości zbliżenia się do powierzchni.

11. Aplikacja na tym etapie ma działać na wszystkich urządzeniach, systemach i przeglądarkach wspierających WebGL. Minimalna liczba klatek na sekundę to 30. Szczegółowe wymagania opisano w rozdziale **4.3 Analiza funkcjonalna**.

4.2 User stories

Jako użytkownik...

#	Chcę...	Żeby...	Kryteria akceptacji
1.	sterować kamerą	obserwować planetę z każdej strony	Użytkownik może w sposób intuicyjny przemieszczać kamerę znajdującą się w pewnej stałej odległości od powierzchni planety.
2.	generować planety ziemio-podobne		Planeta ma zbiorniki wodne oraz lądy o zróżnicowanych rozmiarach. Da się wyróżnić góry, wyżyny i niziny. Powierzchnia pokolorowana jest tak jak mapy topologiczne - nizina na zielono, wyżyna na żółto, góra na brązowo itp.
3.	generować planety pustynne		Powierzchnia planety jest płaska i w kolorze piasku, ale w niektórych miejscach występują góry ciemniejszej barwy.
4.	generować planety lodowe		Powierzchnia planety jest wyraźnie pofałdowana. Kolor powierzchni nie zależy od kształtu.
5.	generować planety lawowe		Powierzchnia planety nie jest znacząco pofałdowana. Na planecie znajduje się sieć rzek magmowych, dzieląca powierzchnię na tafle zastygniętej magmy.
6.	generować gwiazdy		Powierzchnia gwiazdy jest płaska, ale animowana.

4.3 Analiza funkcjonalna

Ze względu na relatywnie niedużą ilość werteksów do wyrenderowania, oczekujemy, że aplikacja będzie działać płynnie (tj. wyświetlając min. 30 klatek/sekundę) na wszystkich urządzeniach, które spełnią poniższe wymagania:

1. System operacyjny:
 - Windows 7 lub nowszy
 - Mac OS 10.6 lub nowszy
 - Linux
 - System operacyjny Chrome
2. Przeglądarka internetowa:
 - Internet Explorer 11 lub nowszy
 - Firefox w wersji 31 lub nowszej
 - Chrome w wersji 31 lub nowszej
 - Safari w wersji 5.1 lub nowszej
 - Opera w wersji 25 lub nowszej
3. Karta graficzna:
 - Intel HD Graphics 5000 lub lepsza
 - NVIDIA GeForce GT 550M lub lepsza
 - AMD Radeon HD 6650M lub lepsza

4.4 Podział prac

Jakub Skąlecki

- Generowanie planet
- Testy jednostkowe

Piotr Leniec

- Generowanie planet
- Ogólna architektura aplikacji

Mateusz Chechliński

- Dokumentacja techniczna i użytkowa
- Plan testów akceptacyjnych

5. Iteracja nr 2

5.1 Cel iteracji

Celem drugiej iteracji jest rozbudowanie stworzonej do tej pory aplikacji o funkcjonalność pozwalającą na dynamiczną zmianę skali poziomu szczegółów (Level of Details). Aplikacja na tym etapie ma mieć następujące cechy:

1. Planeta lub gwiazda jest generowana na podstawie ziarna (seed). Ziarno jest losowane za każdym razem, gdy przeładowywana jest strona.
2. Planeta może należeć do jednego z rodzajów wymienionego projekcie iteracji nr 1.
3. Powierzchnie planet i gwiazd posiadają takie same cechy jak w iteracji nr 1.
4. To czy zostanie wygenerowana gwiazda, czy planeta oraz ewentualnie jakiego rodzaju jest planeta, jest losowe.
5. Planeta lub gwiazda znajduje się w pustej przestrzeni i jest oświetlona równomiernie z każdej strony.
6. Użytkownik może poruszać się swobodnie po przestrzeni kosmicznej, w tym również oddalając się i zbliżając do powierzchni planety.
7. W zależności od odległości powierzchni planety/gwiazdy od obserwatora, dobierany jest poziom szczegółów. Zmiana następuje w czasie rzeczywistym i w sposób niezauważalny dla użytkownika.
8. Aplikacja po tym etapie ma działać na wydajnych komputerach PC. Szczegóły opisano w punkcie 5.3. **Analiza funkcjonalna**

5.2 User stories

W tej iteracji, w porównaniu do poprzedniej, dodano jedną User Story:

Jako użytkownik...

#	Chcę...	Żeby...	Kryteria akceptacji
1.	sterować kamerą	obserwować planetę z bliska	<p>Użytkownik może zbliżyć się do powierzchni planety. Wówczas aplikacja zwiększy poziom szczegółów.</p> <p>Gdy użytkownik oddali się, poziom szczegółów zostanie zmniejszony.</p>

5.3 Analiza funkcjonalna

Ze względu na większą złożoność obliczeniową, zmodyfikowaliśmy minimalne wymagania sprzętowe. Po zakończeniu tego etapu, aplikacja powinna działać z prędkością min. 30 kl./sek. na urządzeniach spełniających poniższe kryteria:

1. System operacyjny:
 - Windows 7 lub nowszy
 - Mac OS 10.6 lub nowszy
 - Linux
2. Przeglądarka internetowa:
 - Firefox w wersji 31 lub nowszej
 - Chrome w wersji 31 lub nowszej
3. Karta graficzna:
 - NVIDIA GeForce GTX 650M lub lepsza
 - AMD Radeon HD 7770 lub lepsza

5.4 Podział prac

Jakub Skąlecki

- Algorytm Level of Details

Piotr Leniec

- Algorytm generowania tekstur

Mateusz Chechliński

- Dokumentacja techniczna i użytkowa

6. Iteracja nr 3

6.1 Cel iteracji

Celem trzeciej iteracji jest wykorzystanie dotychczasowych rozwiązań do generowania i prezentowania fragmentu przestrzeni kosmicznej, zawierającego co najmniej kilkadziesiąt planet i gwiazd.

Jako że cały projekt koncentruje się na możliwie najwydajniejszym wykorzystaniu mocy obliczeniowej karty graficznej, nie zaś na symulacji kosmosu wg praw fizyki, odległości między obiektami powinny zostać zmniejszone, nie należy również przykładać uwagi do proporcji rozmiarów planet i gwiazd.

Aplikacja na tym etapie ma mieć następujące cechy:

1. Galaktyka jest generowana na podstawie ziarna. Ziarno jest losowane za każdym razem, gdy przeładowywana jest strona.
2. Galaktyka wypełniona jest planetami i gwiazdami stworzonymi w poprzednich iteracjach.
3. Planety i gwiazdy renderowane w taki sposób, że są oświetlone z każdej strony i nie rzucają cienia.
4. Użytkownik może płynnie i szybko zmieniać odległość od powierzchni planety/gwiazdy, a także w intuicyjny sposób zmieniać obserwowane obiekty.
5. Użytkownik może przemieszczać kamerę wokół obserwowanego obiektu.
6. W zależności od odległości powierzchni planety/gwiazdy od obserwatora, dobierany jest poziom szczegółów. Zmiana następuje w czasie rzeczywistym i w sposób niezauważalny dla użytkownika.
7. Aplikacja po tym etapie ma działać na typowych komputerach PC. Szczegóły opisano w punkcie 6.3. **Analiza funkcjonalna**

6.2 User stories

W tej iteracji, w porównaniu do poprzedniej, zmodyfikowaliśmy User Stories opisujące sposób w jaki użytkownik porusza się po przestrzeni kosmicznej oraz związane z liczbą planet i gwiazd:

Jako użytkownik...

#	Chcę...	Żeby...	Kryteria akceptacji
1.	generować fragment kosmosu		<p>Aplikacja generuje i wyświetla co najmniej kilkadziesiąt planet i gwiazd jednocześnie.</p> <p>Użytkownik może zobaczyć wszystkie obiekty znajdujące się w otoczeniu obserwowanej planety.</p> <p>Odległości pomiędzy obiektami nie muszą być realistyczne.</p>
2.	regulować odległość	płynnie przechodzić od poziomu powierzchni planety do poziomu galaktyki	<p>Użytkownik może w sposób intuicyjny zmienić odległość od planety / gwiazdy w sposób płynny.</p> <p>Zmiana poziomu szczegółów odbywa się w czasie rzeczywistym.</p> <p>Niezależnie od odległości od powierzchni, aplikacja renderuje około 60 klatek / sek.</p>
3.	przełączać się między planetami	sprawnie nawigować na całej dostępnej przestrzeni kosmicznej	<p>Użytkownik może jednym kliknięciem zmienić obiekt wokół którego się porusza.</p> <p>Zmiana następuje natychmiastowo, bez konieczności oczekiwania na wczytanie czy załadowanie następnego obiektu.</p>
4.	przemieszczać kamerę wokół planety / gwiazdy	obserwować obiekty z każdej strony	<p>Użytkownik może w intuicyjny sposób zmienić kąt z jakiego obserwuje wybrany obiekt.</p> <p>Zmiana następuje w czasie rzeczywistym, przy zachowaniu stałego poziomu FPS</p>

6.3 Analiza funkcjonalna

Jako że projekt jest skoncentrowany na wykorzystaniu maksimum możliwości, jakie WebGL daje programistom, postawiliśmy sobie za cel by aplikacja działała płynnie nawet na starszych i wydajnych urządzeniach, w tym również na komputerach wyposażonych w zintegrowane karty graficzne. W związku z tym na tym etapie ponownie obniżyliśmy wymagania sprzętowe:

1. System operacyjny:
 - Windows 7 lub nowszy
 - Mac OS 10.6 lub nowszy
 - Linux
2. Przeglądarka internetowa:

- Firefox w wersji 31 lub nowszej
- Chrome w wersji 31 lub nowszej

3. Karta graficzna:

- Intel HD Graphics 5000 lub lepsza
- NVIDIA GeForce GT 550M lub lepsza
- AMD Radeon HD 6650M lub lepsza

6.4 Podział prac

Jakub Skąlecki

- Optymalizacja algorytmu Level of Details
- Udoskonalenie algorytmu generowania planet

Piotr Leniec

- Generowanie galaktyki

Mateusz Chechliński

- Generowanie powierzchni różnych rodzajów planet i gwiazd
- Dokumentacja techniczna i użytkowa

7. Iteracja nr 4

7.1 Cel iteracji

Celem ostatniej iteracji jest wzbogacenie projektu o efekty związane z oświetleniem. Położona w centralnym punkcie każdego systemu gwiazda jest źródłem światła dla otaczających ją planet.

Każda planeta powinna być z jednej strony oświetlona, a z drugiej zupełnie zacieniona, chyba że na planecie znajdują się źródła światła (jak np. rzeki lawy na planecie lawowej).

Cieniowanie planet nie może się przełożyć na widoczny spadek wydajności aplikacji. Wobec tego nie ulegają zmianie wymagania sprzętowe opisane w iteracji nr 3.

7.2 Analiza funkcjonalna

Wymagania sprzętowe w tej iteracji nie uległy zmianie względem iteracji nr 3.

7.3 Podział prac

Jakub Skąlecki

- Model oświetlenia i cieniowania

Piotr Leniec

- Naprawa wykrytych dotychczas błędów

Mateusz Chechliński

- Dokumentacja techniczna i użytkowa