Politechnika Gdańska

WYDZIAŁ ELEKTRONIKI
TELEKOMUNIKACJI I INFORMATYKI

Dokumentacja projektu dyplomowego inżynierskiego

Edytor do proceduralnego generowania modeli drzew

Łukasz Odzioba 119415
Mariusz Okrój 119416

Opiekun pracy:
dr inż. Michał Małafiejski

Katedra opiekuna pracy:
Katedra Algorytmów i Modelowania Systemów

Streszczenie dokumentu:
Dokument zawiera opis narzędzia do łatwego tworzenia trójwymiarowych modeli drzew w oparciu o zadany przez użytkownika zbiór parametrów opisujących oczekiwany wygląd modelu.

Gdańsk, 2011

OŚWIADCZENIE

Oświadczam, że niniejszą pracę dyplomową wykonałem samodzielnie. Wszystkie informacje umieszczone w pracy uzyskane ze źródeł pisanych oraz informacje ustne pochodzące od innych osób zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami.

.....
podpis dyplomanta

Spis treści

Rozdział 1

Wstęp

1. Cel pracy

Celem ninejszej pracy inżynierskiej jest stworzenie silnika dynamiki brył sztywnych działającego w czasie rzeczywistym, który mógłby być wykorzystywany do symulacji fizycznych na potrzeby gier komputerowych. Oznacza to między innymi, że efekty pracy silnika powinny raczej wyglądać wiarygodnie, niż skupiać się na dokładnym odwzorowaniu zjawisk fizycznych. Dlatego wystarczający jest uproszczony model fizyczny. Z drugiej strony silnik musi radzić sobie z symulacją dużej liczby obiektów równocześnie w czasie rzeczywistym, czyli znając stan świata dla chwili T, musi wyliczyć stan świata dla chwili $T + \Delta T$ w czasie rzeczywistym nie dłuższym niż ΔT.

Dodatkowym celem pracy jest poszerzenie wiedzy o silnikach fizycznych, poprzez zbadanie możliwości silników obecnie istniejących, oraz poprzez porównanie ich z właściwościami silnika tworzonego w ramach tej pracy inżynierskiej. Jest to o tyle ważne, że ninejsza praca proponuje innowacyjny sposób obliczania właściwości fizycznych oraz wykrywania kolizji między obiektami oparty o wypełnienie obiektów Kisielkami[1].

Celem pobocznym pracy jest również stworzenie przykładowej aplikacji lub prostej gry prezentującej możliwości silnika, aby umożliwić przyszłym użytkownikom, oraz innym osobom zainteresowanym naszym silnikiem, szybkie i łatwe zapoznanie się ze sposobem wykorzystania silnika oraz oferowaną przez niego funkcjonalnością.

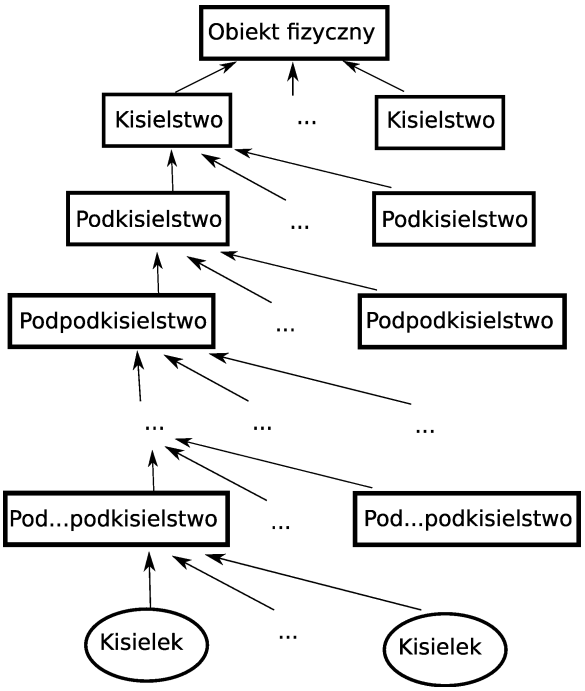
[1] pojęcie wyjaśnione w następnej sekcji — 'Definicje pojęć'

2. Definicje pojęć

- Kisielek — cząstka dyskretna, przybliżająca masę i objętość swojego najbliższego otoczenia. Pojedynczy obiekt fizyczny może się składać z dowolnie dużej, ale skończonej, liczby Kisielków. Kisielki pojedynczego obiektu nie przemieszczają się względem siebie (wg. definicji bryły sztywnej). Kolizje obiektów są rozpatrywane jako kolizje pomiędzy parami Kisielków danych obiektów.

- Kisielstwo — podgrupa Kisielków pojedynczego obiektu. Obiekt fizyczny może być podzielony na wiele rozłącznych Kisielstw. Kisielki są grupowane w celu przyspieszenia wykrywania kolizji, poprzez wstępne rozpatrywanie kolizji między grupami Kisielków — Kisielstwami.

- Podkisielstwo — podgrupa Kisielków należących do pojedynczego Kisielstwa. Kisielki tworzą hierarchiczną strukturę, którą można przedstawić jako drzewo:



- Impuls — momentalna zmiana pędu i momentu pędu obiektu np. na skutek kolizji. Jest to pojęcie stworzone w celu uniknięcia obliczania siły dążącej do nieskończoności przy czasie dążącym do zera, które by wynikło z założenia, że czas kolizji jest pomijalnie krótki.

- Wszechkisiel — zarządca, pojedynczy na całą symulację obiekt, u którego użytkownik rejestruje informacje pomiędzy którymi obiektami mają być wykrywane kolizje, a pomiędzy którymi nie. Odpowiada również za stałe globalne (np. przyspieszenie grawitacyjne, długość kroku symulacji).

3. Zadania do wykonania i podział pracy

Implementacja

Kinematyka	zagadnienia związane z położeniem i orientacją obiektu oraz ich pochodnymi, w tym całkowanie numeryczne mające na celu wyliczenie kolejnego stanu obiektu	JR
Kinetyka	obliczanie m, I, $\vec{F}$, $\vec{\tau}$, oraz ich wpływu na dynamikę obiektu	JR
Relacje między obiektami	powiązania między obiektami, np. sprężyny, nacisk statyczny, obiekty sklejone ze sobą, zawiasy	JR
Obsługa kolizji	metodą sprężyn-amortyzatorów	JR
Rozpadanie się obiektów	reakcja niektórych obiektów na silne kolizje, skutkująca rozpadem na mniejsze obiekty	JR
Wypełnianie Kisielkami	algorytm wypełniający zadaną bryłę geometryczną Kisielkami i tworzący ich hierarchię	KB
Algorytm podziału	dzielący obiekt na Kisielstwa i kolejne poziomy, w taki sposób, aby zoptymalizować późniejsze wykrywanie kolizji	KB
Wykrywanie kolizji	pomiędzy kulami opisanymi na Kisielstwach, oraz pomiędzy kolejnymi poziomami hierarchii, aż do poszczególnych Kisielków	KB
Obsługa kolizji	metodą impulsów	KB
Zarządca (Wszechkisiel)	stworzenie obiektu zarządzającego wykrywaniem kolizji (grupy kolizyjne), oraz globalnymi parametrami silnika (np. grawitacja)	KB

Inne

Tworzenie dokumentacji	JR
Testy wydajnościowe	JR
Przykładowa aplikacja demonstracyjna	JR
Opracowanie algorytmów	KB

Wizualizacja wypełnienia Kisielkami	KB

4. Harmonogram

Faza analizy i projektowania

Zapoznanie się z istniejącymi silnikami fizycznymi	1.10.2011 – 15.10.2011
Przeczytanie "Fizyki dla programistów gier"[1]	1.10.2011 – 1.11.2011
Opracowanie algorytmów wykrywania kolizji, obsługi kolizji, wypełniania geometrii Kisielkami, całkowania numerycznego, oraz rozpadania się obiektów	1.10.2011 – 1.11.2011
Implementacja prototypów na podstawie podręcznika[1]	15.10.2011 – 1.11.2011
Zaprojektowanie interfejsów poprzez które użytkownik będzie z silnika korzystał	15.10.2011 – 1.11.2011
Zaprojektowanie interfejsów komunikacji pomiędzy poszczególnymi modułami silnika	15.10.2011 – 1.11.2011
Zaprojektowanie architektury silnika	15.10.2011 – 7.11.2011

Faza implementacji

Implementacja kinematyki, kinetyki i wypełniania Kisielkami	1.11.2011 – 15.11.2011
Implementacja wykrywania kolizji, obsługi kolizji, oraz relacji między obiektami	15.11.2011 – 1.12.2011
Implementacja aplikacji demonstracyjnej	15.11.2011 – 7.12.2011
Implementacja rozpadania się obiektów, oraz zarządcy (Wszechkiśla)	1.12.2011 – 7.12.2011

Faza zakończeniowa

Testy wydajnościowe i jakościowe	15.11.2011 – 7.12.2011
Opracowanie ostatecznej dokumentacji projektu i instrukcji obsługi	15.11.2011 – 7.12.2011
Przygotowanie produktu do przekazania (spakowanie jako biblioteki, wraz z zależnościami, oraz dołączoną dokumentacją)	1.12.2011 – 7.12.2011

Rozdział 2

Wstęp teoretyczny

1. Historia i stan obecny dziedziny

2. Algorytm kolonizacyjny

3. Tworzenie geometrii modelu

Przykład kodu źródłowego:

```
@Override
public String toString() {
    return "x: " + x + " y: " + y + " z: " + z;
}
```

4. Teksturowanie

Rozdział 3

Specyfikacja wymagań

http://entropy.echelon.pl/miguel/print.php?id=11

1. **Ograniczenia dziedziny tematu**
2. **Use cases albo user stories**

Rozdział 4

Dokumentacja

1. **Diagramy klas**
2. **Opis formatu .obj i procedury eksportu**
3. **Opis interfejsu uzytkownika**
4. **Repozytorium git, buildowanie projektu, potrebne biblioteki**

Rozdział 5

Problemy i rozwiązania, testy

1. **Problemy i rozwiazania**
2. **Testy wydajnościowe i jakościowe**
3. **Przykladowa galeria z parametrami uzytymi do generowania**

Rozdział 6

Podsumowanie i wnioski

powinno zawierać opis osiągnięć z rozbiciem na poszczególnych członków zespołu, możliwości dalszego rozwoju i inne uwagi. LO: - jakich kurwa osiagniec? juz samo to, ze sie udalo zrobic cos dzialajacego na czas to jest osiagniecie

1. Możliwosci rozwoju

inne obslugiwane formaty kosci i fizyka drzewa optymalizacja metod generowania drzewa dodanie nowych modyfikatorow bloby i galezie anatomiczne korzen proceduralne liscie i tekstura kory kwiaty owoce na drzewie generowanie kilku modeli jednoczesnie lepsze wyswietlanie w edycji ustawianie lisci do swiatla cofnij powtorz przy modyfikacji kilkustopniowe generowanie w srodku drzewa zwykle nie ma lisci inne algorytmy generowania - uwzgledniajace np oswietlenie, wplyw wiatru animacja wzrostu drzewa

Rozdział 7

Załączniki

Zrzuty ekranu, płyta z kodem, programem i dokumentacją, instrukcje obsługi, łotewer

Bibliografia

[1] A. Runions, B. Lane, P. Prusinkiewicz: *Modeling Trees with a Space Colonization Algorithm*. Eurographics Workshop on Natural Phenomena 2007