POLITECHNIKA WROCŁAWSKA

AUTOMATYKA I ROBOTYKA, ARR, W4

Wizualizacja Danych Sensorycznych

SPRAWOZDANIE OKRESOWE NR 1/3
- WSTĘPNE REZULTATY -

Wizualizacja rozkładu ciśnienia cieczy na podstawie symulacji komputerowej

Autorzy: Balawender Adam Kwieciński Krzysztof Prowadzący: Dr inż. Bogdan Kreczmer

23 kwietnia 2015

Spis treści

1	Opis projektu		
2	SPH		
3	Plan pracy3.1 Podział obowiązków3.2 Harmonogram3.3 Kamienie milowe3.4 Diagram Gantta	2	
4	Funkcjonalności aplikacji		
5	Interfejs graficzny		
6	Diagram klas		
7	Przepływ sterowania		

1 Opis projektu

Zgodnie z tematem projektu zajmiemy się komputerową symulacją zachowania cieczy oraz wizualizacją jej stanu i rozkładu ciśnienia w zbiorniku z płynem.

Symulacja będzie obejmowała ruch cieczy w przekroju 2D wybranego naczynia. Ciecz zostanie przedstawiona na płaszczyźnie jako zbiór oddziaływujących ze sobą cząsteczek. Postaramy się, żeby jej zachowanie było możliwie zbliżone do rzeczywistego. Ruch płynu zostanie zamodelowany metodą numeryczną SPH (smoothed particle hydrodynamics - wygładzona hydrodynamika cząstek). Pozwoli to na realistyczne odwzorowanie zachowania cieczy. Możliwe będzie badanie cieczy o różnych parametrach, dlatego też modelowane będą jej właściwości fizyczne: gęstość i lepkość. Dodatkowo mierzone będzie ciśnienie cieczy i zostanie ono zwizualizowane jako odcień koloru płynu. Im będzie on ciemniejszy, tym wyższe ciśnienie będzie odzwierciedlał.

2 SPH

Metoda zakłada reprezentację cieczy przez zbiór cząstek N_i o jednakowej masie \mathbf{m} (charakteryzowanych w momencie \mathbf{i} pozycją $\mathbf{r_i}$, prędkością $\mathbf{v_i}$ oraz gęstością ρ_i), które reagują na siebie na odległość \mathbf{h} . Korzystając z wyprowadzeń opisanych w artykule [2] przyjęliśmy wzór na gestość w kroku \mathbf{i} :

$$\rho_i = \frac{4m}{\pi h^8} \sum_{i \in N_i} (h^2 - r^2)^3.$$

Przyspieszenie będziemy natomiast wyznaczać ze wzoru:

$$\mathbf{a_i} = \frac{1}{\rho_i} \sum_{j \in N_i} \mathbf{f}_{ij}^{interact} + \mathbf{g},$$

gdzie

$$\mathbf{f}_{ij}^{interact} = \frac{m_j}{\pi h^4 \rho_j} (1 - q_{ij}) \left[15k(\rho_i + \rho_j - 2\rho_0) \frac{1 - q_{ij}}{q_{ij}} \mathbf{r}_{ij} - 40\mu \mathbf{v}_{ij} \right],$$

gdzie $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ i $q_{ij} = ||\mathbf{r}_{ij}||/h$, ρ_0 - wyjściowa gęstość, k - współczynnik sprężystości objętościowej, μ - lepkość, g - wektor grawitacji.

3 Plan pracy

3.1 Podział obowiązków

Projekt zakłada powiązanie symulacji numerycznej (back-end) z aplikacją prezentującą wyniki w formie graficznej (front-end). Za pierwszą z ww. części odpowiedzialny będzie Adam Balawender, za drugą Krzysztof Kwieciński. Obie części powinny mieć możliwość niezależnego uruchomienia, co ułatwi ich testowanie we wstępnych etapach oraz ocenę w końcowym etapie projektu.

3.2 Harmonogram

Termin rozpoczęcia projektu to 9.03.2015, natomiast termin zakończenia realizacji projektu przewidziany jest na 11.06.2015. Oznacza to, że okres prac nad projektem wynosi w przybliżeniu XIII tygodni.

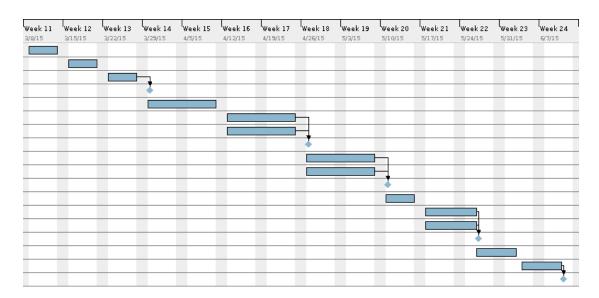
Do tej pory udało się nam całkowicie zrealizować zadania z tygodni I - IV z poprzedniego harmonogramu, [1]. Na dzień dzisiejszy projekt jest realizowany już przez VI tygodni. Zaktualizowany harmonogram został przedstawiony poniżej.

Tydzień	Adam	Krzysztof	
I	Opis projektu		
II	Przegląd bibliotek Qt, szkic GUI		
III	Zapoznanie się z metodą SPH (Smoothed Particle Hydrodynamics)		
IV, V	Ustalenie struktur danych oraz API modułów		
VI, VII	Implementacja klas zbiornika oraz cząsteczek cieczy	Stworzenie statycznej wizualizacji zbiornika	
VIII, IX	Dodanie wizualizacji położenia czą- stek cieczy	Implementacja metod uaktualniania położenia cząsteczek	
X	Analiza błędów działania programu i skorygowanie ich		
XI	Wyznaczanie ciśnienia w punkach zbiornika	Wizualizacja ciśnienia w punktach	
XII	Weryfikacja projektu z założeniami i odpowiednie modyfikacje programu		
XIII	Napisanie raportu końcowego		

3.3 Kamienie milowe

- K1. Przeanalizowanie artykułów na temat SPH i zapoznanie się z tą metodą
- K2. Zaimplementowanie struktur danych, modelu cieczy i relacji między cząsteczkami
- K3. Wizualizacja symulowanego stanu cieczy
- K4. Wizualizacja ciśnienia w poszczególnych punktach zbiornika
- K5. Skończona dokumentacja

3.4 Diagram Gantta



Rysunek 1: Diagram Gantta

4 Funkcjonalności aplikacji

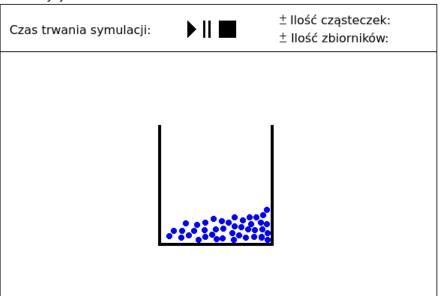
Najistotniejszymi funkcjonalnościami aplikacji będą:

- symulacja zachowania cieczy w zależności od zadanych warunków początkowych,
- możliwość przedefiniowania parametrów cieczy (gęstości, lepkości),
- możliwość obserwacji wyniku symulacji (położenia cząsteczek i rozkładu ciśnień).

5 Interfejs graficzny

Szkic planowanego interfejsu graficznego aplikacji przedstawiony jest na rysunku 2.

Plik Edycja Widok Pomoc



Rysunek 2: Szkic interfejsu graficznego

6 Diagram klas

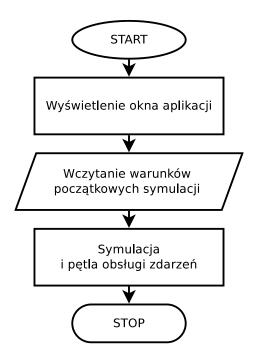
Diagram klas przedstawiony jest na rysunku 3.

```
Wektor
                                                                                                      Zbiornik
_x: double
_y: double
                                                                               szerokosc: double
                                                                               _wysokosc: double
                                                                               _ilosc_czasteczek: int = 0
+dodai(in a:const Wektor&.in b:const Wektor&): Wektor const
+odejmij(in a:const Wektor&,in b:const Wektor&): Wektor const
                                                                                _czasteczki: list<Czasteczka>
+pomnoz(in a:const Wektor&,in b:const Wektor&): Wektor const
                                                                               +Zbiornik(in szerokosc:double,in wysokosc:double,
                                                                               in ilosc_czasteczek:int=0)
+maksymalnaPojemnosc(): double const
+Wektor(x:double,y:double)
+Wektor()
+x(): double&
                                                                               pojemnoscAktualna(): double const
+x(): double const
+y(): double&
                                                                              +dodajCzasteczke(): void
+usunCzasteczke(): Czasteczka
+y(): double const
                                                                               +wypiszZawartosc(): void const
                                                                              +szerokosc(): double&
+szerokosc(): double const
                      Czasteczka
                                                                              +wysokosc(): double&
+wysokosc(): double const
+ilosc_czasteczek(): int const
-_promien: double
-_gestosc: double
-_polozenie: Wektor
                                                                               +czasteczki(): list<Czasteczka>&
 _kolor: string
+stworzNCzasteczek(in ilosc:int=0): list
                                                                                                          PanelSterowania
+Czasteczka(in promien:double,in lepkosc:double,
              in gestosc:double,in polozenie:Wektor
                                                                                                     -_plik_testowy: string
              in kolor:string)
                                                                                                     -_ilosc_zbiornikow: int = 1
+Czasteczka()
                                                                                                     _zbiorniki: list<Zbiornik>
+promien(): double&
                                                                                                    +play(): void
+promien(): double const
+polozenie(): Wektor&
                                                                                                     +pause(): void
                                                                                                    +stop(): void
+dodajZbiornik(): void
polozenie(): Wektor const
+kolor(): string&
+kolor(): string const
                                                                                                    +usunZbiornik(): Zbiornik
+plik_testowy(): string&
+plik_testowy(): string const
                                       Symulacja
                                                                                                    +ilosc_zbiornikow(): int const
+zbiorniki(): list<Zbiornik>&
- iloscCzastek: int
-<u>_</u>masaCzastki: float
 _rozmiarCzastki: float
- deltaT: float
-_gestosc: float
 ___wspolczynnikSprezystosci: float
 lenkosc: float
__przyspieszenieGrawitacyjne: float
+obiekt: PanelSterowania
+stworzNCzasteczek(in iloscCzastek:int,in masaCzastki:float,
                      in rozmiarCzastki:float,
                      in deltaT:float,in gestosc:float,
                      in wspolczynnikSprezystosci:float,
                      in lepkosc:float,in przyspieszenieGrawitacyjne:float): list
```

Rysunek 3: Diagram klas dla interfejsu graficznego

7 Przepływ sterowania

Diagram przepływu sterowania przedstawiony jest na rysunku 4.



Rysunek 4: Diagram przepływu sterowania dla interfejsu graficznego

Literatura

- [1] A. Balawender and K. Kwieciński. Wizualizacj rozkładu ciśnienia cieczy na podstawie symulacji komputerowej (założenia projektowe), marzec 2015.
- [2] D. Bindel. http://www.cs.cornell.edu/bindel/class/cs5220-f11/code/sph-derive.pdf.
- [3] T. Harada, S. Koshizuka, and Y. Kawaguchi. http://inf.ufrgs.br/cgi2007/cd_cgi/papers/harada.pdf.
- [4] Qt. http://doc.qt.io/.
- [5] Qt Quick 2D Render. http://doc.qt.io/QtQuick2DRenderer/index.html.