

POLITECHNIKA WROCŁAWSKA

AUTOMATYKA I ROBOTYKA, ARR, W4

WIZUALIZACJA DANYCH SENSORYCZNYCH

SPRAWOZDANIE OKRESOWE Nr 2/3
- REZULTATY PRAWIE KOŃCOWE -

Wizualizacja rozkładu ciśnienia cieczy na podstawie symulacji komputerowej

Autorzy:

BALAWENDER Adam
KWIECIŃSKI Krzysztof

Prowadzący:

Dr inż. Bogdan KREZMER

15 maja 2015

Spis treści

1	Opis projektu	1
2	Plan pracy	1
2.1	Podział obowiązków	1
2.2	Harmonogram	1
2.3	Kamienie milowe	2
2.4	Diagram Gantta	3
3	SPH	3
4	Funkcjonalności aplikacji	4
5	Interfejs graficzny	4
6	Przepływ sterowania	5
7	Diagram klas	6
8	Dokumentacja	7

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.

Aplikacja zostanie napisana w języku C++, przy użyciu biblioteki Qt.

2 Plan pracy

2.1 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.

2.2 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 zrealizować zadania z tygodni I - X z założonego wcześniej harmonogramu, [1]. Na dzień dzisiejszy prace nad projektem trwają już przez dziesięć tygodni, zatem projekt wykonywany jest w terminie.

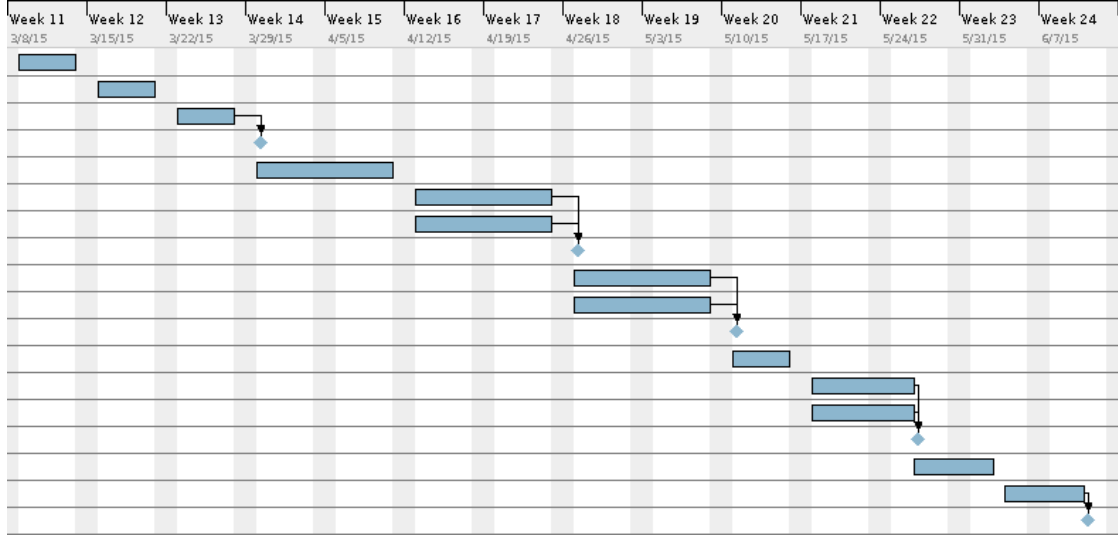
Aktualny harmonogram przedstawiony znajduje się poniżej.

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 cząstek cieczy	Implementacja metod uaktualniania położenia cząsteczek
X	Analiza błędów działania programu i skorygowanie ich	
XI	Wyznaczanie ciśnienia w punktach zbiornika	Wizualizacja ciśnienia w punktach
XII	Weryfikacja projektu z założeniami i odpowiednie modyfikacje programu	
XIII	Napisanie raportu końcowego	

2.3 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

2.4 Diagram Gantta



Rysunek 1: Diagram Gantta

3 SPH

Metoda zakłada reprezentację cieczy przez zbiór cząstek \mathbf{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 gęstość w kroku \mathbf{i} :

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

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, \mathbf{g} - wektor grawitacji.

4 Funkcjonalności aplikacji

Najistotniejsze funkcjonalności aplikacji to:

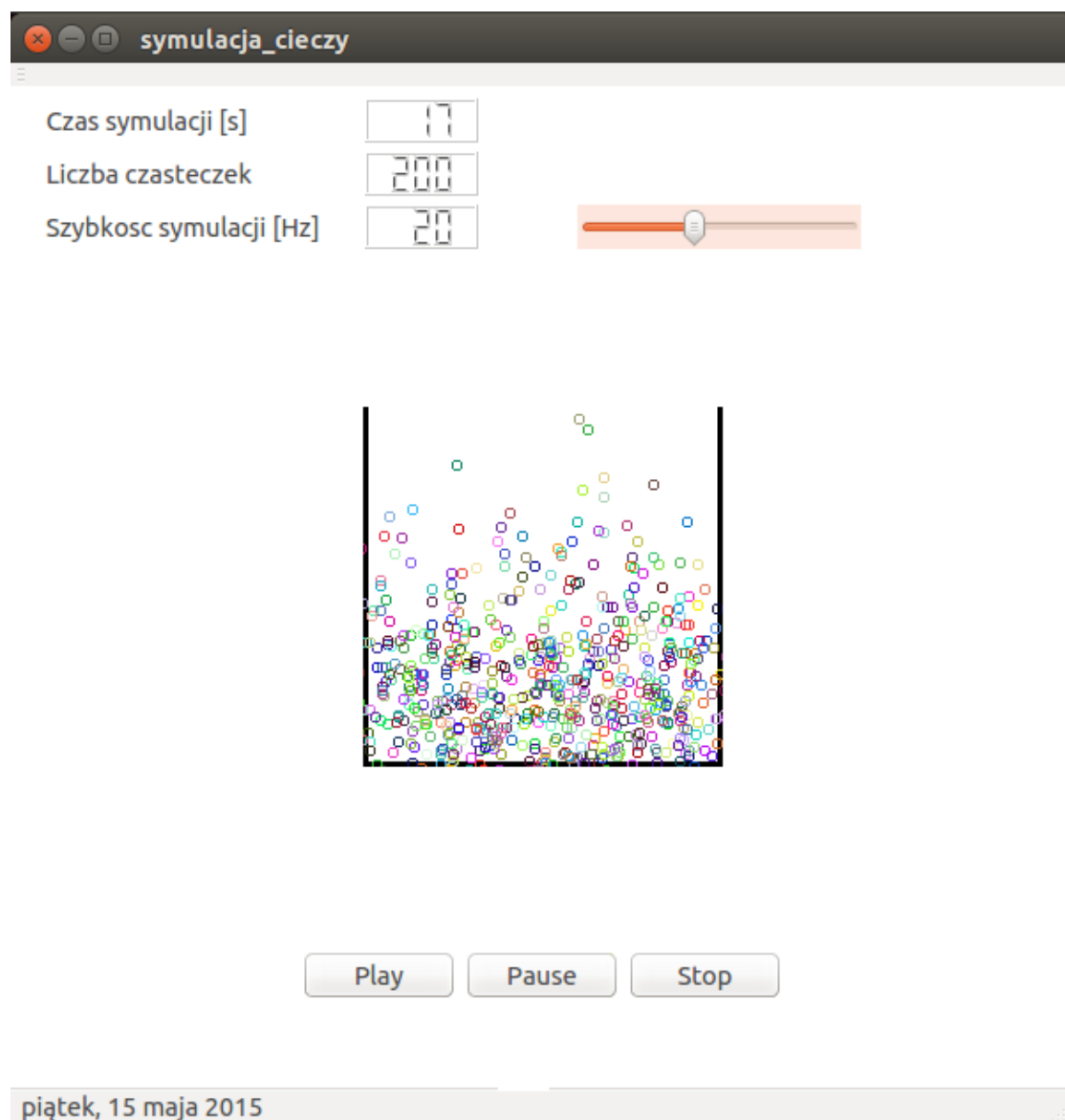
- symulacja zachowania cieczy w zależności od zadanych warunków początkowych,
- możliwość przededefiniowania parametrów cieczy (gęstości, lepkości),
- możliwość obserwacji wyniku symulacji (położenia cząsteczek i rozkładu ciśnień).

Do tej pory udało się nam już zasymulować zachowanie cieczy. Test poprawności napisanego algorytmu realizującego metodę *SPH* przeprowadziliśmy w Pythonie (wizualizacja - rysunek 2), a także w C++ (dane liczbowe). Ponadto programowo dostępna jest zmiana wszystkich założonych parametrów cząsteczki cieczy. Podczas działania aplikacji w jej okienku obserwować można chaotycznie poruszające się cząsteczki cieczy. Rysunek 3 przedstawia przykładowy zrzut ekranu działającej aplikacji.

Rysunek 2: Symulacja cieczy w Pythonie

5 Interfejs graficzny

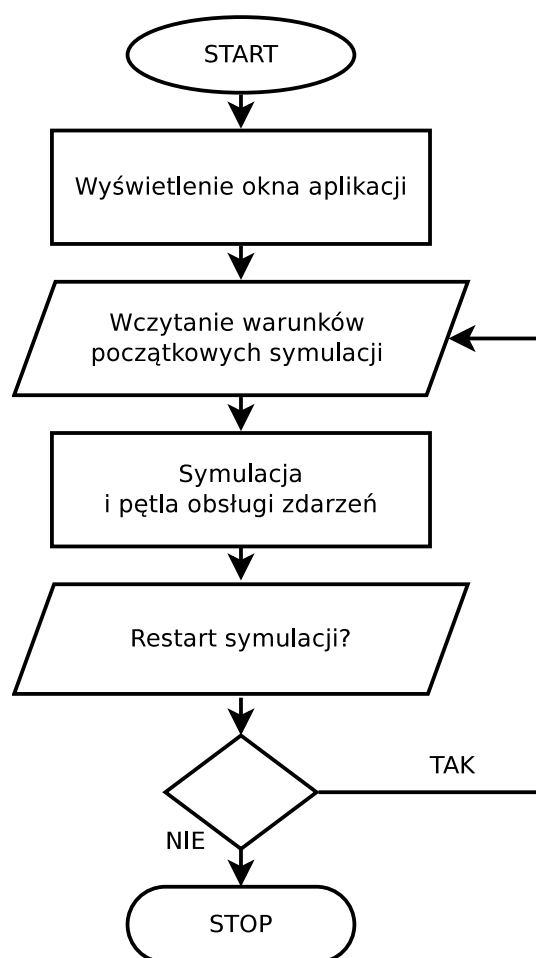
Aktualny wygląd interfejsu graficznego aplikacji przedstawiony jest na rysunku 3.



Rysunek 3: Interfejs graficzny aplikacji

6 Przepływ sterowania

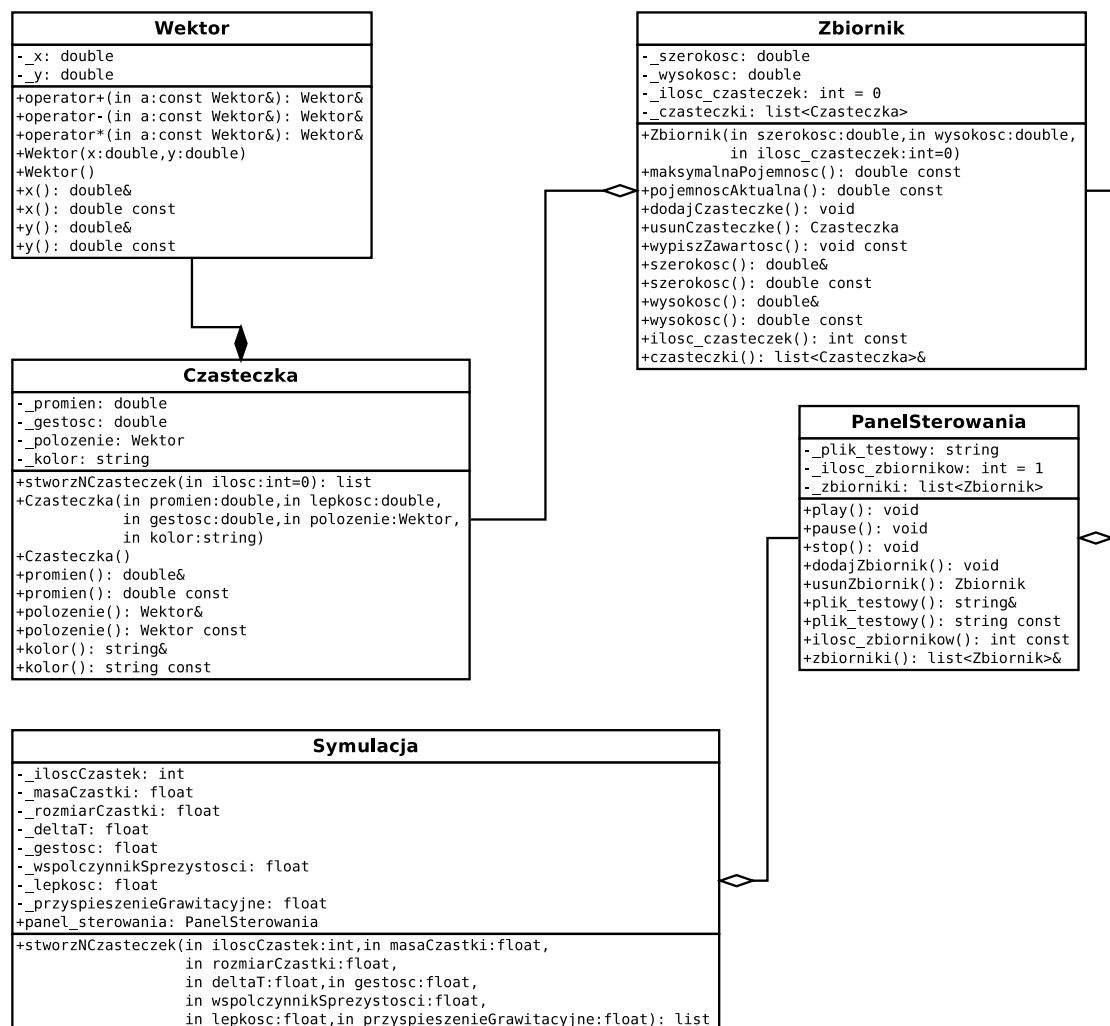
Diagram przepływu sterowania przedstawiony jest na rysunku 4.



Rysunek 4: Diagram przepływu sterowania dla interfejsu graficznego

7 Diagram klas

Założony diagram klas przedstawiony jest na rysunku 5.



Rysunek 5: Diagram klas dla interfejsu graficznego

8 Dokumentacja

Powstające oprogramowanie jest na bieżąco dokumentowane. Dokumentacja powstaje z zastosowaniem systemu generowania dokumentacji Doxygen, [6].

Literatura

- [1] A. Balawender and K. Kwieciński. Wizualizacj rozkładu ciśnienia cieczy na podstawie symulacji komputerowej (wstępne rezultaty), kwiecień 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] D. Solin. *Poznaj programowanie przy użyciu biblioteki Qt w 24 godziny*. Infoland, Warszawa, 2001.
- [6] D. van Heesch. <http://www.doxygen.org/>.