

POLITECHNIKA WROCŁAWSKA

AUTOMATYKA I ROBOTYKA, ARR, W4

WIZUALIZACJA DANYCH SENSORYCZNYCH

SPRAWOZDANIE OKRESOWE Nr 3/3
- REZULTATY KOŃCOWE -

Wizualizacja rozkładu ciśnienia cieczy na podstawie symulacji komputerowej

Autorzy:

BALAWENDER Adam
KWIECIŃSKI Krzysztof

Prowadzący:

Dr inż. Bogdan KREZMER

11 czerwca 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 Gantt'a	3
3	SPH	3
4	Funkcjonalności aplikacji	4
5	Interfejs graficzny	4
6	Diagram klas	4
7	Przepływ sterowania	7
8	Aplikacja	7
9	Dokumentacja	10
10	Wnioski	10

1 Opis projektu

Projekt dotyczy komputerowej symulacji zachowania cieczy oraz wizualizacji jej stanu i rozkładu ciśnienia w zbiorniku z płynem.

Symulacja obejmuje ruch cieczy w przekroju 2D wybranego naczynia. Ciecz jest przedstawiona na płaszczyźnie jako zbiór oddziałujących ze sobą cząsteczek. Jej zachowanie jest możliwie zbliżone do rzeczywistego dzięki modelowaniu ruchu płynu przy pomocy metody numerycznej SPH (*smoothed particle hydrodynamics* - *wygładzona hydrodynamika cząstek*). W programie modelowane są właściwości fizyczne cieczy: gęstość i lepkość. Dlatego też można badać zachowania płynów o różnych parametrach. Dodatkowo prowadzony jest pomiar ciśnienia cieczy. Ciśnienie wizualizowane jest jako odcień koloru płynu. Im jest on ciemniejszy, tym wyższe ciśnienie odzwierciedla.

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

2 Plan pracy

2.1 Podział obowiązków

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

2.2 Harmonogram

Projekt realizowano w okresie 9.03.2015 - 11.06.2015. Oznacza to, że okres prac nad projektem wyniósł w przybliżeniu XIII tygodni.

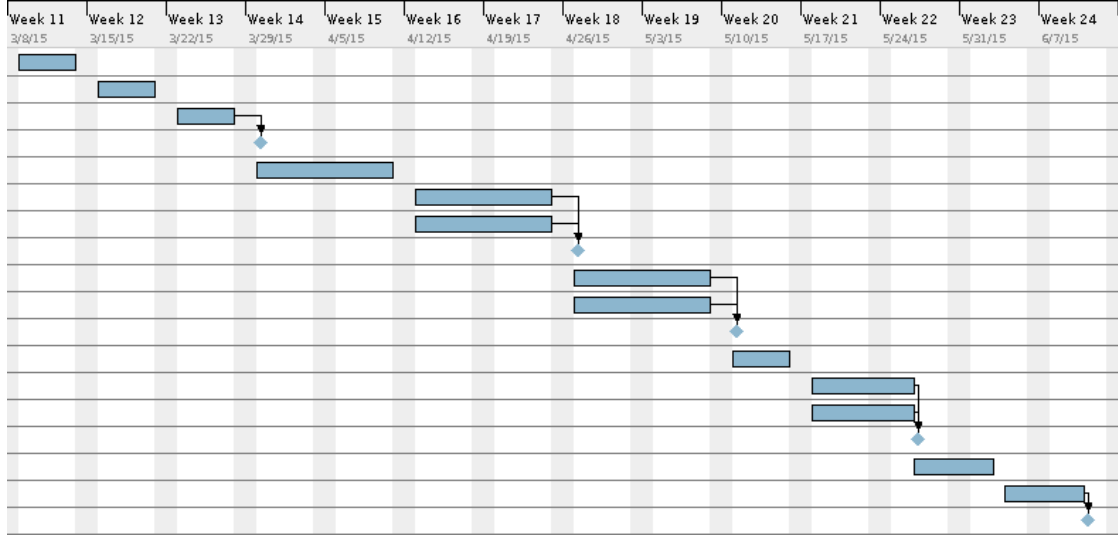
Cały projekt wykonany został w terminowo. Zrealizowano wszystkie zaplanowane zadania z tygodni I - XIII z założonego wcześniej harmonogramu, [1].

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	Wizualizacja ciśnienia w punktach	Wyznaczanie ciśnienia w punktach zbiornika
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] przyjęto 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 wyznaczano 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,
- symulacja zachowania cieczy przy poruszaniu zbiornikiem,
- symulacja zachowania cieczy na Ziemi lub w stanie nieważkości,
- programowa możliwość przededefiniowania parametrów cieczy (gęstości, lepkości),
- zmiana szybkości symulacji,
- możliwość obserwacji wyniku symulacji (położenia cząsteczek i rozkładu ciśnień).

Aplikacja umożliwia zasymulowanie zachowania cieczy zarówno od zadanych warunków początkowych, jak i poprzez interakcję użytkownika z programem (poruszanie zbiornikiem za pomocą *slidera*). Symulacja odzwierciedla zachowanie cząsteczek na Ziemi lub w Kosmosie. Programowo dostępna jest zmiana wszystkich założonych parametrów cząsteczki cieczy. Podczas działania aplikacji w jej okienku obserwować można komputerowo zamodelowany ruch płynu wraz z rozkładem panujących w nim ciśnień. Symulacja może być zatrzymywana i ponownie uruchamiana, a jej szybkość zmieniana. Rysunek 2 przedstawia przykładowy zrzut ekranu działającej aplikacji.

5 Interfejs graficzny

Wygląd interfejsu graficznego aplikacji przedstawiony jest na rysunku 2.

W centralnej części aplikacji widoczny jest jej główny element, czyli zbiornik z cząsteczkami. Pod nim znajduje się *slider* pozwalający na przechylenie zbiornika.

Cząsteczki oraz okienko aplikacji są częściowo przezroczyste, co pozwala na obserwowanie elementów znajdujących za cieczą.

Użytkownik za pomocą trzech przycisków umiejscowionych na dole ekranu może sterować symulacją, a mianowicie ją: uruchomić, zamrozić lub zatrzymać.

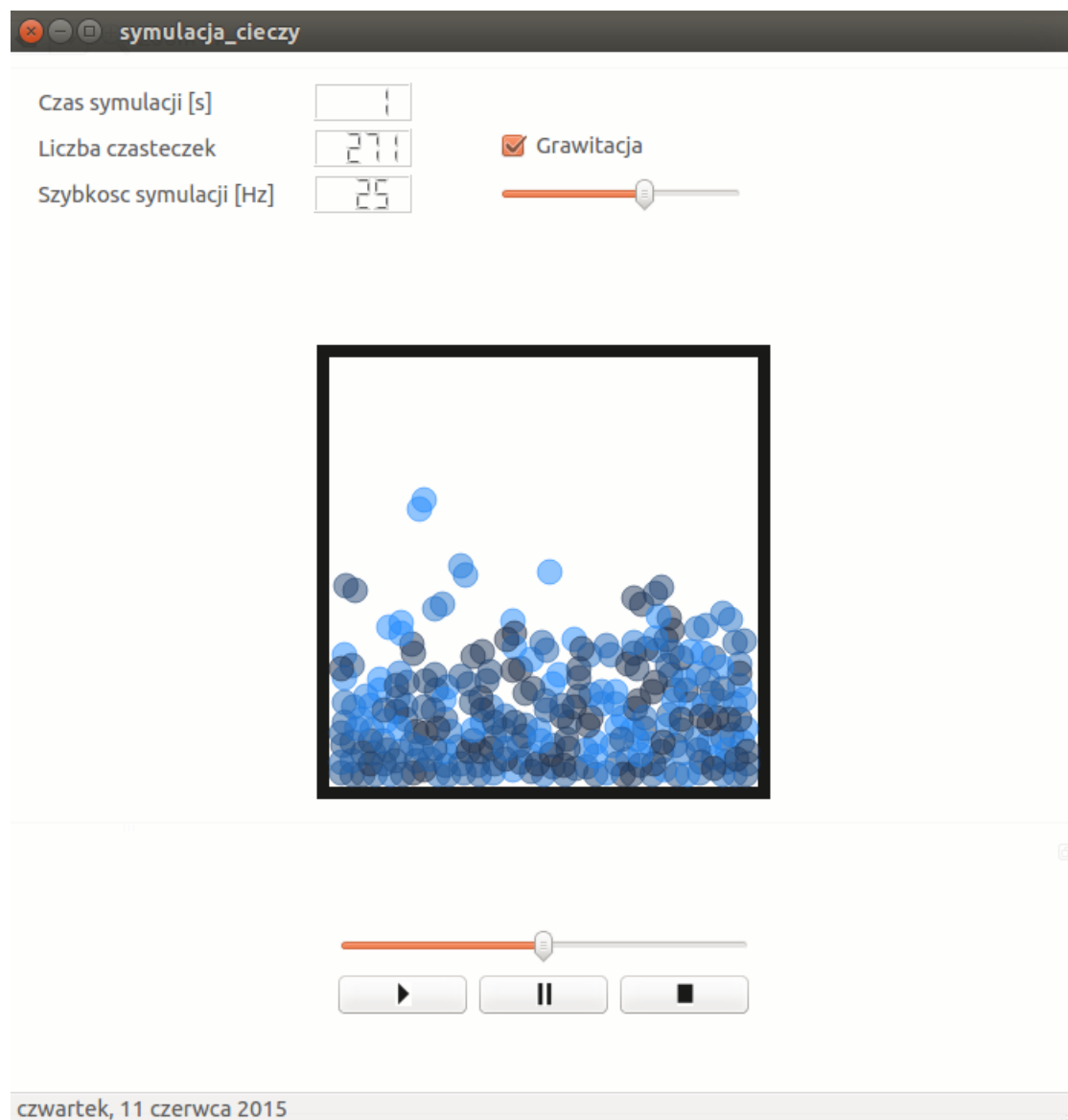
W lewej górnej części okienka znajdują się elementy informacyjne pozwalające śledzić parametry symulacji: jej czas trwania, symulowaną liczbę cząsteczek oraz szybkość odświeżania wizualizacji. Na prawo od nich umiejscowione są elementy umożliwiające interakcję użytkownika z aplikacją: *slider* pozwalający na zmianę szybkości symulacji oraz pole typu *checkbox* zmieniające wektor grawitacji.

Na widocznej w dole ekranu belce statusowej wyświetlana jest aktualna data. W paśmie menu dostępna jest opcja zamknięcia aplikacji.

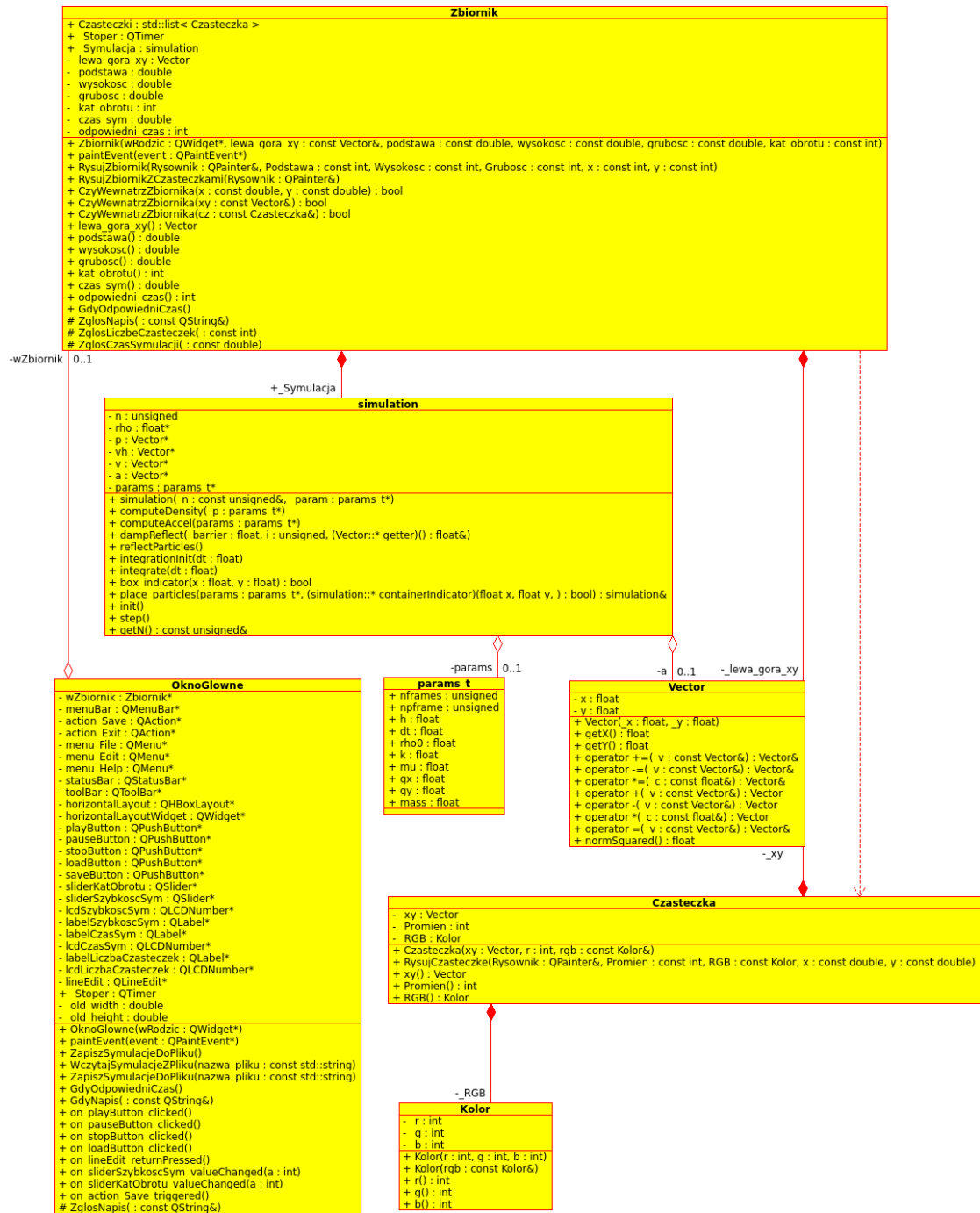
Rozkład elementów interfejsu graficznego jest odpowiednio modyfikowany przy zmianie wymiarów okienka.

6 Diagram klas

Diagram klas, wygenerowany programem Umbrello, przedstawiony jest na rysunku 3.



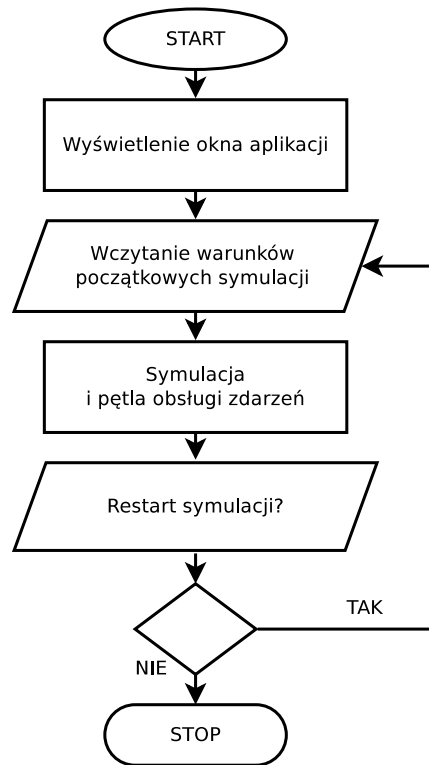
Rysunek 2: Interfejs graficzny aplikacji



Rysunek 3: Diagram klas aplikacji

7 Przepływ sterowania

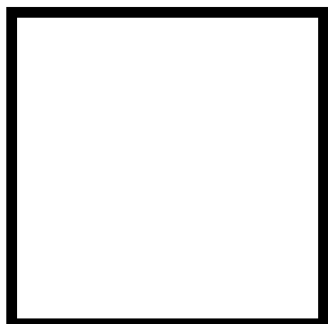
Diagram przepływu sterowania przedstawiony jest na rysunku 4.



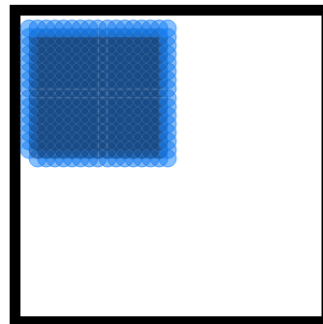
Rysunek 4: Diagram przepływu sterowania

8 Aplikacja

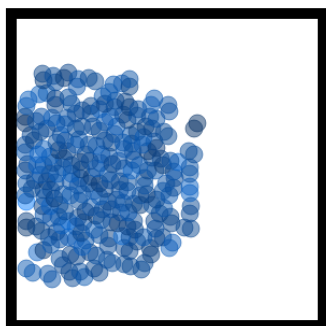
Na rysunkach 5 - 12 znajdują się przykłady uzyskiwanych efektów pracy aplikacji. Prze-testowane zostały jej wszystkie założone funkcjonalności.



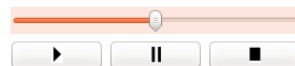
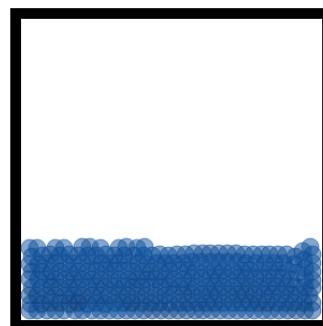
Rysunek 5: Pusty zbiornik



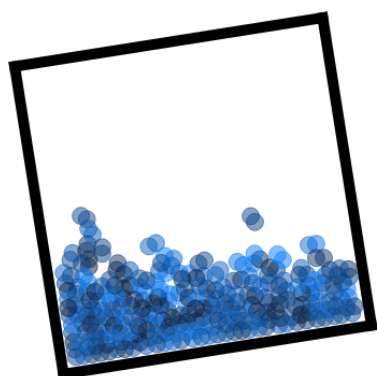
Rysunek 6: Wybrany stan początkowy



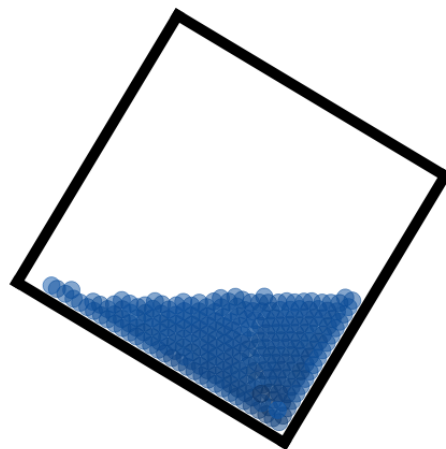
Rysunek 7: Symulacja



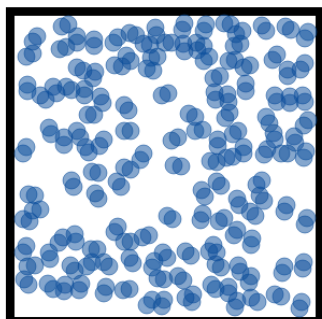
Rysunek 8: Ustabilizowana ciecz



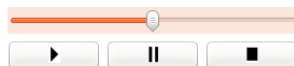
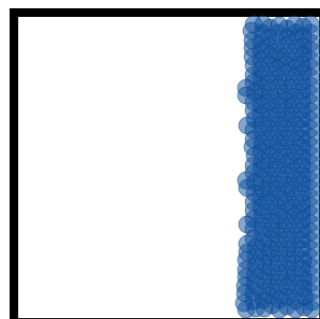
Rysunek 9: Obracanie zbiornika



Rysunek 10: Ustabilizowana ciecz



Rysunek 11: Stan nieważkości



Rysunek 12: Stan nieważkości

9 Dokumentacja

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

Dokumentacja aplikacji wygenerowana w postaci html dostarczona jest w ramach materiałów dodatkowych.

10 Wnioski

Pisanie aplikacji przy użyciu biblioteki Qt pozwala na osiągnięcie dobrych efektów przy stosunkowo małym nakładzie pracy. Po zaznajomieniu się z ww. biblioteką oraz aplikacją designer programy można pisać w efektywny sposób. Duża ilość gotowych elementów umożliwia projektowanie aplikacji bez potrzeby ponownej implementacji podstawowych składników każdego programu graficznego.

Dzięki hierarchii elementów graficznych możliwe jest ich łatwe wykorzystanie w aplikacji. Ponadto mechanizm dziedziczenia dostarcza wielu wspólnych atrybutów i metod dla różnych elementów, co znacząco poprawia komfort pisania kodu programu.

W Qt używany jest mechanizm slotów i sygnałów. Pozwala on na wywoływanie metod w odpowiednich momentach w odpowiedzi na jakieś wydarzenia. Dzięki temu możliwa jest interakcja użytkownika z aplikacją, a także synchronizacja obiektów w programie.

Zastosowanie metody `QObject::tr` pozwoli na ewentualną internacjonalizację aplikacji.

Metoda SPH w sposób realistyczny modeluje zachowanie cząsteczek i pozwala na ich sparametryzowanie. Okazała się być ona bardzo dobrym wyborem dla komputerowej realizacji symulacji cieczy.

W każdym projekcie bardzo ważna jest faza analizy i planowania. Dzięki niej możliwe jest dobre zaplanowanie harmonogramu działań, co pozwala na późniejsze realizowanie projektu zgodnie z założeniami.

Praca w dwuosobowym zespole jest bardzo dobrym pomysłem. Umożliwia to podział zadań i pozwala na wzajemną pomoc, a także umożliwia naukę od drugiej osoby.

Dokumentowanie kodu na bieżąco pozwala na łatwiejszą pracę nad projektem. Dodatkowo dobra dokumentacja znacząco poprawia czytelność kodu i umożliwia postronnym osobom na jego lepsze zrozumienie i wykorzystanie. Za dobry przykład może posłużyć dokumentacja Qt, [4], która jest bardzo szczegółowa. Dzięki temu mogliśmy w miarę bezproblemowo nauczyć się pisania aplikacji graficznych z wykorzystaniem tej biblioteki.

Literatura

- [1] A. Balawender and K. Kwieciński. *Wizualizacja 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/>.