Zaawansowane programowanie w języku JAVA

Sprawozdanie z projektu

Korsarze-PC

Kamil Gasik

Maciej Ogonowski

E6C1S4

Github: https://github.com/MaciejOgonowski1/Korsarze-PC

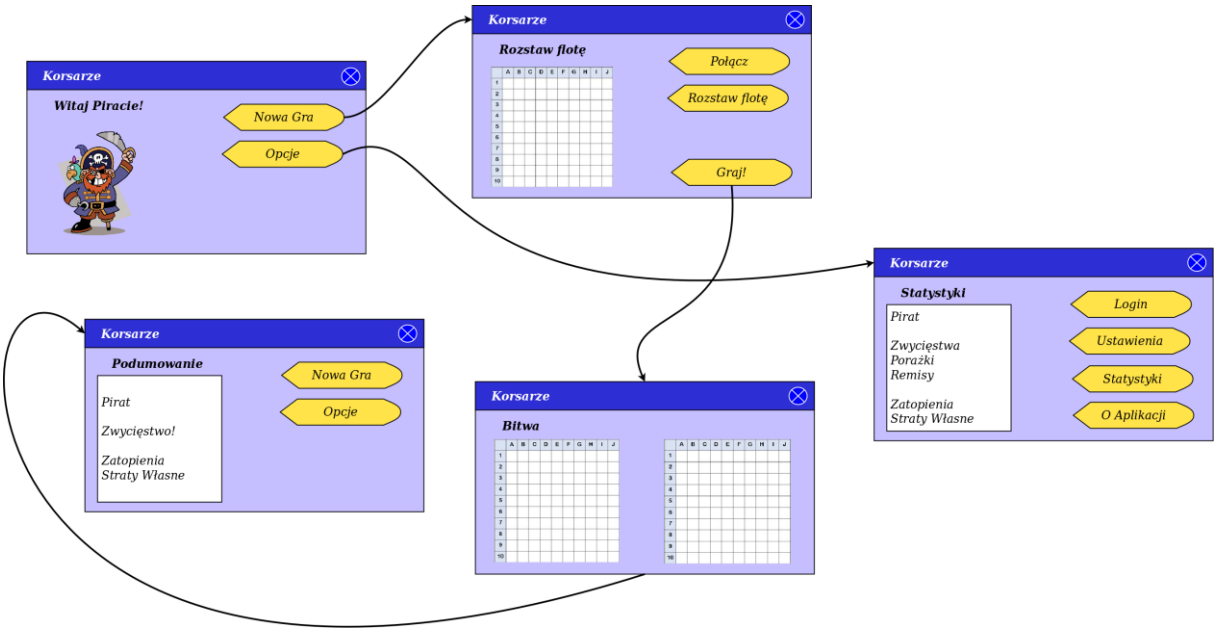
1. Opis aplikacji

Korsarze-PC to klasyczna gra w statki dla dwóch osób. Polega na zatopieniu statków przeciwnika zanim zatopi on nasze statki. Gra odbywa się na planszy 10x10 pól, osobnej dla każdego z graczy. Każdy z nich ustawia na planszy statki zajmujące łączne 20 pól. Ustawienie może być dowolne lub zgodne z ustaleniami grających.

Projekt aplikacji zakładał możliwość gry w statki:

- Rozstawienie statków
- Połączenie między graczami
- Bitwę
- Dodatkową funkcjonalność (personalizację(login), statystyki, ustawienia)

Poniżej zaprezentowano projekt koncepcyjny gry. Ostateczny wygląd został zmodyfikowany i uproszczony.



Rys. 1. Projekt koncepcyjny

2. Podział zadań i harmonogram

Podział zadań:

Maciej Ogonowski

- Komunikacja między graczami
- GUI

Kamil Gasik

- Algorytm gry
- GUI

Projekt zakładał wykonywanie powyższych zadań na kolejne spotkania seminaryjne zgodnie z harmonogramem:

- 19.10.2017 - Rozpoczęcie prac koncepcyjnych
- 16.11.2017 - GUI
- 14.12.2017 - Algorytm gry, komunikacja między graczami
- 24.01.2017 - Ukończenie prac nad algorytmem gry oraz komunikacją między graczami, (dodatkowa funkcjonalność)

3. Realzacja projektu

Została zrealizowana podstawowa funkcjonalność aplikacji oraz interfejs do funkcjonalności dodatkowej (Opcje).

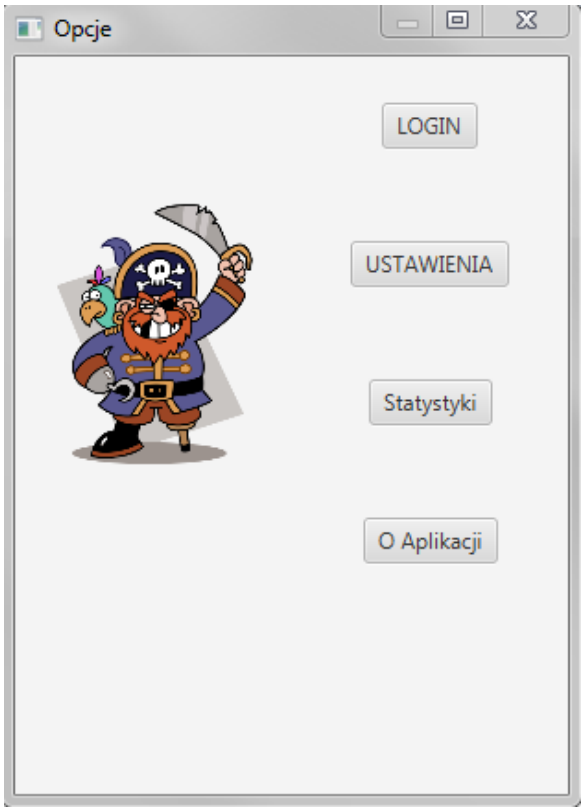
Interfejs

Do realizacji Interfejsu graficznego wykorzystano bibliotekę *javafx.* Elementy statyczne interfejsu graficznego zostały wykonane przy pomocy programu *SceneBuilder*. Do wygenerowania planszy został użyty obiekt *GridPane* o wymiarach 10x10 i wykorzystane metody:

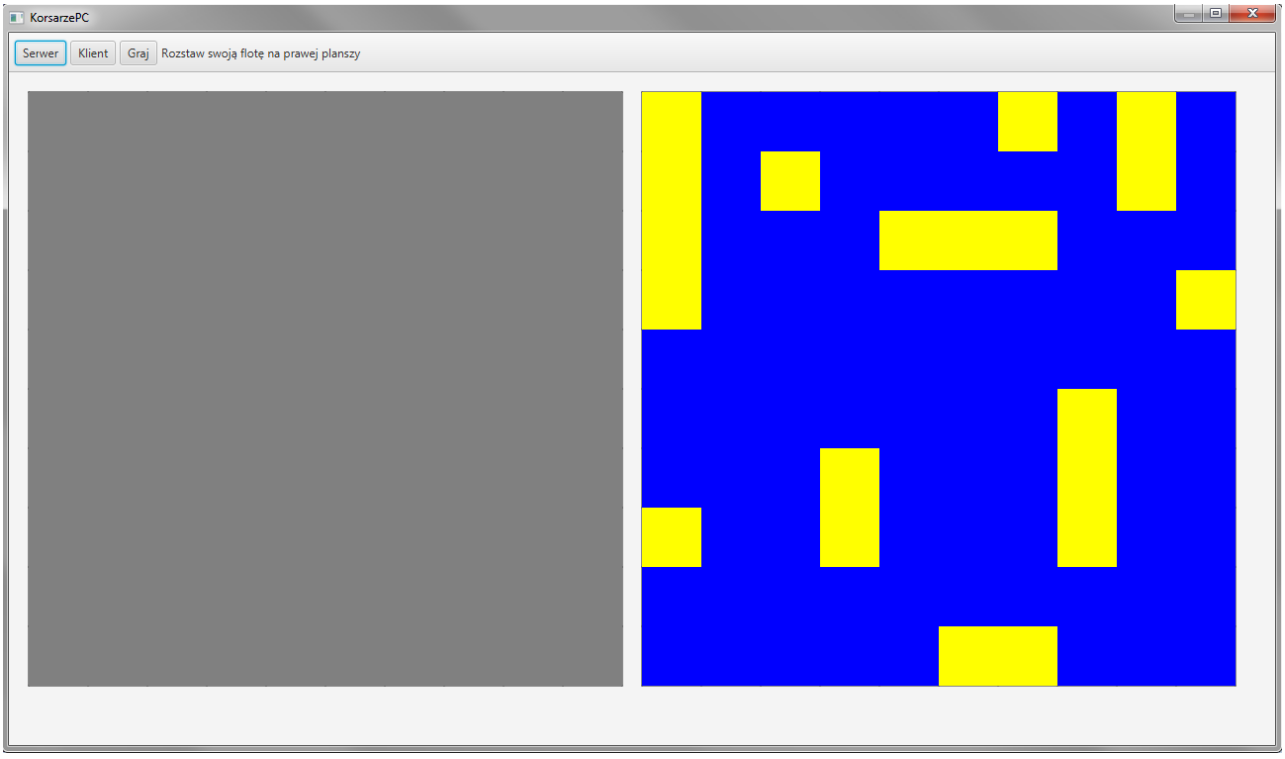
On Mouse Clicked
On Mouse Entered

służące do generacji i obsługi interfejsu planszy. *GridPane* został wypełniony obiektami *Pane* w odpowiednim kolorze. Niestety nie udało się zastosować obiektów *ImageView* wypełnionych odpowiednimi grafikami, z tego względu pola o tym samym kolorze zlewają się w całość. Pola planszy wypełniane są po najechaniu kursorem na jedną z map. Nad planszami znalazło się etykieta tekstowa zawierająca podstawowe komunikaty.
Interfejs ustawiania statków został połączony razem z interfejsem bitwy.
Poniżej znajdują się grafiki przedstawiające interfejs gry:

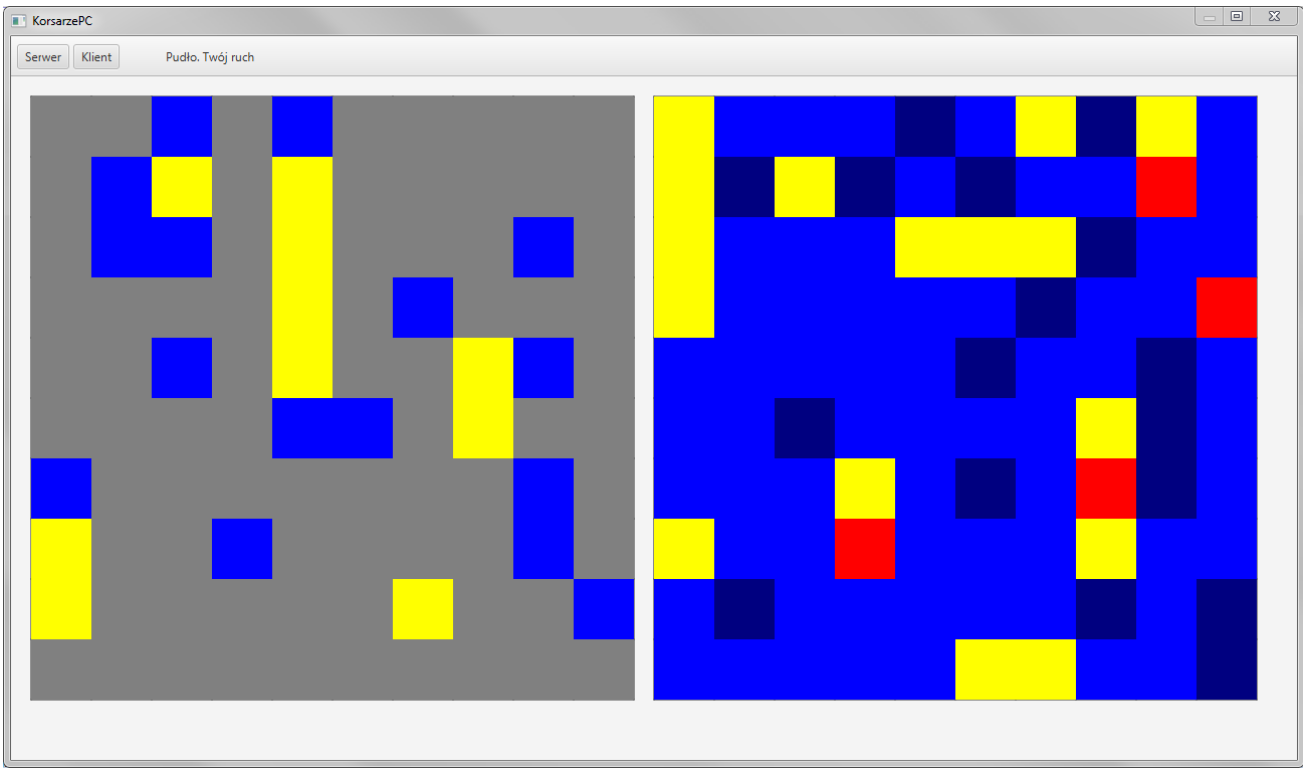
Rys. 2. Menu główne



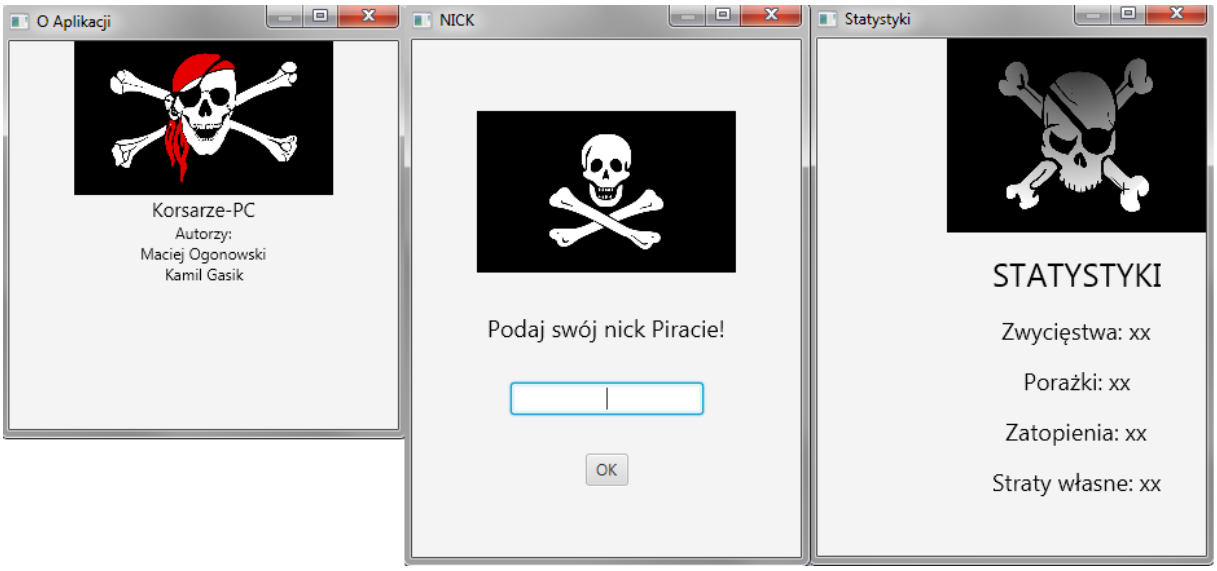
Rys. 3. Menu opcji



Rys. 4. Pole gry(ustawianie statków)



Rys. 5. Pole gry (bitwa)



Rys. 6. Pozostałe zrzuty ekranu(O Aplikacji, Login, Statystyki)

Komunikacja

Komunikacja została zrealizowana z użyciem biblioteki *java.net*. Komunikacja odbywa się z użyciem localhost'a. Przed rozpoczęciem gry należy określić, które z urządzeń pełni rolę klienta/serwera. Komunikacja między dwoma urządzeniami prowadzona jest w formie zapytań i odpowiedzi. Zapytanie składa się z wysłania dwóch komunikatów zawierających współrzędne zapisane w postaci liczb całkowitych ze znakiem (typ int). Odpowiedź składa się z jednego komunikatu zawierającego liczbę typu int równą 1 – gdy pod podanymi współrzędnymi znajdował się statek, lub 0 – gdy spudłowano. Za przesyłane komunikatów i czekanie na odpowiedź odpowiada algorytm gry. Komunikacja znajduje się w pliku *GrajController.java*.

Algorytm gry

Algorytm gry znajduje się w pliku *Controller.java* razem z obsługą interfejsu graficznego odpowiedzialnego za wyświetlanie i obsługę plansz. Utworzona została także klasa *Board.java* do przechowywania parametrów planszy.

Realizacja gry na jednym ekranie odbywa się dzięki zastosowaniu zmiennej *gameState*, określającej aktualny etap gry i przypisane do niego akcje:

- 0 – przygotowanie do gry – możliwe jest ustawienie statków na własnej planszy. Statki umieszcza się poprzez kliknięcie wybranego pola oraz usuwa przez ponowne kliknięcie. Plansza przeciwnika i przyciski klient/serwer nie są w tym trybie obsługiwane
- 1 – gotowość do połączenia – możliwy jest wybór klienta lub serwera
- 20 – bitwa – ruch gracza – możliwe jest zaznaczenie pola przeciwnika, które ma zostać zaatakowane. Sprawdzane jest, czy dane pole nie zostało wybrane już wcześniej, po trafieniu wybierane jest kolejne pole, po pudle wykonywana jest tura przeciwnika
- 21 –bitwa – ruch przeciwnika – czekanie na ruch przeciwnika, strzela on do momentu spudłowania, wtedy tura przechodzi na gracza
- 3 – koniec bitwy – możliwość zamknięcia aplikacji

Przejście pomiędzy powyższymi stanami jest możliwe w momencie spełnienia określonych warunków:

- Z 0 do 1 – po kliknięciu Graj, gdy na planszy ustawiono dokładnie 20 statków, przycisk zostaje ukryty
- Z 1 do 20 –po połączeniu się jako klient
- Z 1 do 21 – po połączeniu się jako serwer
- Z 20 do 21 – po nietrafieniu statku przeciwnika
- Z 21 do 20 – po nieudanym ataku przeciwnika
- Z 20 lub 21 do 3 – automatycznie, gdy wszystkie statki jednego z graczy zostaną zatopione

Wszystkie akcje podejmowane w czasie gry są dozwolone w odpowiednim czasie, jeżeli dana akcja nie jest możliwa wyświetlany jest stosowny komunikat w *messageLabel* u góry ekranu razem z odpowiednimi sugestiami.

4. Podsumowanie

Podstawowa funkcjonalność została zrealizowana. Należałoby poprawić klika elementów:

- Dodać linie oddzielające pola na map dla uzyskania wyższej czytelności
- Poprawić działanie interfejsu – w czasie oczekiwania na połączenie nie są rysowane trafienia – są uzupełniane dopiero w kolejnej turze, aplikacje może się zawieszać do momentu otrzymania spodziewanego komunikatu, potem działa normalnie
- Dodać dodatkową funkcjonalność