

Przedmiot:	Imię i Nazwisko:	Grupa laboratoryjna:
Programowanie Obiektowe 2	Kacper Kałuża	2ID11A

# Temat: Milionerzy Świętokrzyskiego

## 1. Wstęp

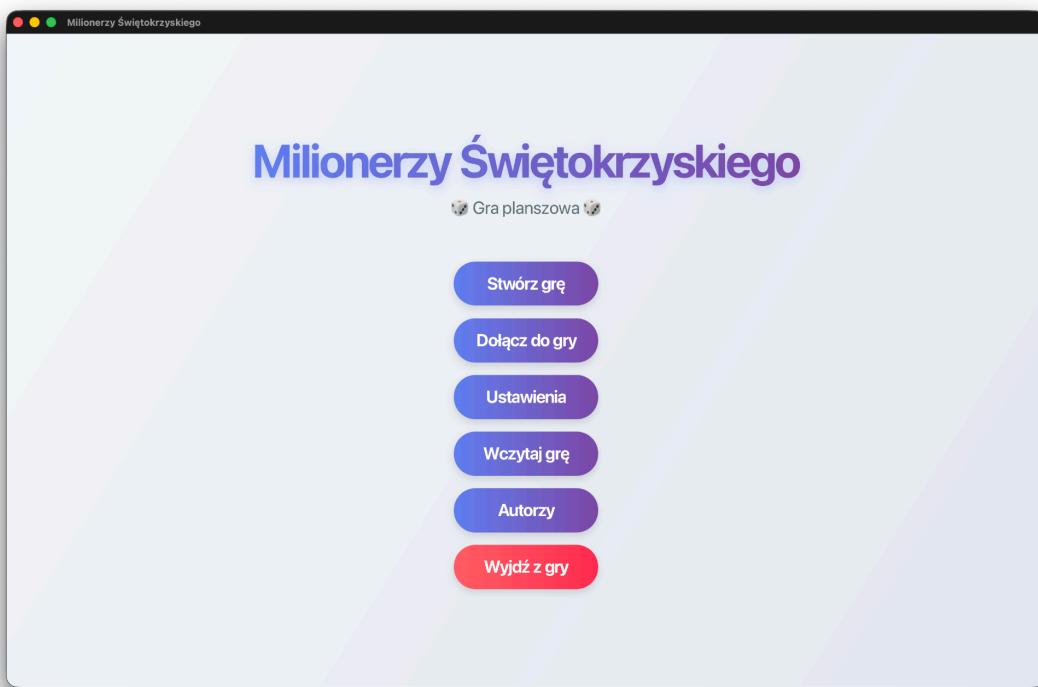
Projekt "Milionerzy Świętokrzyskiego" to gra planszowa typu Monopoly, województwa świętokrzyskiego. Aplikacja została napisana w języku Java (wersja 25) z wykorzystaniem biblioteki JavaFX do obsługi interfejsu graficznego oraz silnika sieciowego opartego na gniazdach TCP. Gra umożliwia rozgrywkę sieciową (Host-Klient).

## 2. Uruchomienie gry

Aby uruchomić grę, należy skompilować projekt przy użyciu Maven (`mvn clean javafx:run`) lub uruchomić klasę główną `com.kaluzaploetecka.milionerzy.Launcher`.

### 3. Menu Głównego

Po uruchomieniu aplikacji użytkownikowi ukazuje się Menu Głównego.



Zrzut Ekranu 1 - Widok Menu głównego

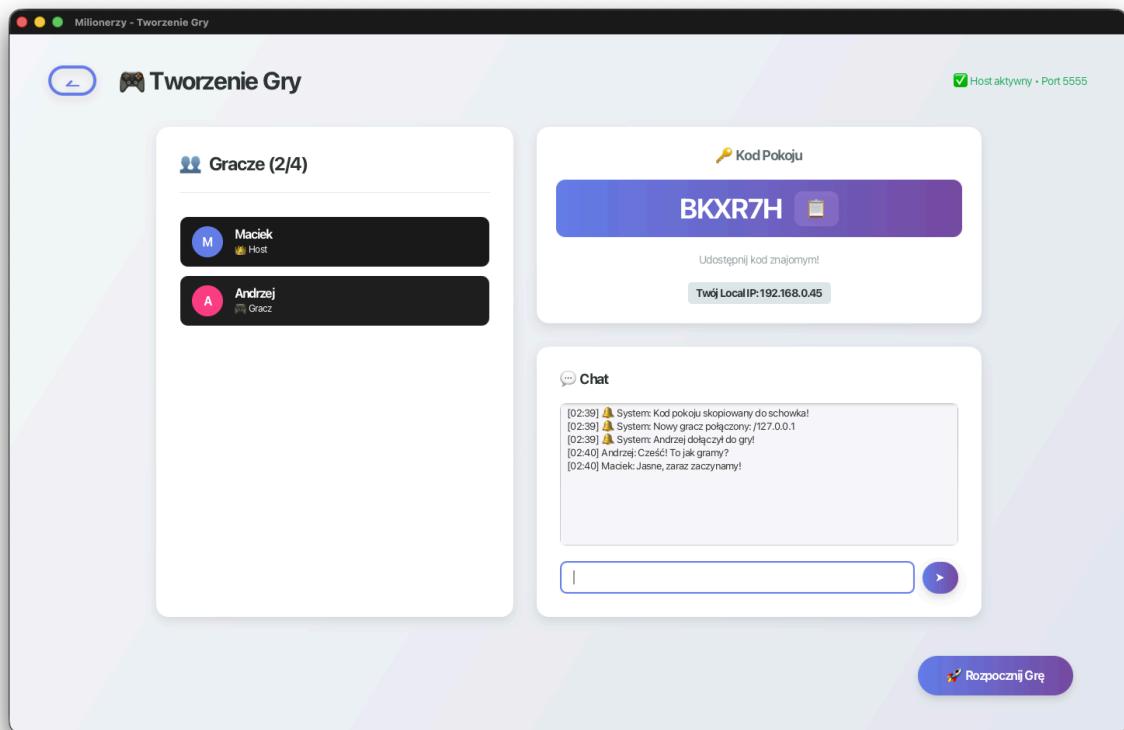
#### Opis elementów ekranu:

- **Stwórz Grę:** Uruchamia tryb Hosta (Serwera). Gracz wpisuje swoją nazwę i czeka na innych graczy.
- **Dołącz do gry:** Pozwala połączyć się z istniejącą grą. Wymaga podania:
  - Nazwy gracza.
  - Kod Pokoju
  - Adresu IP Hosta (domyślnie `localhost` dla gry na jednym komputerze).
- **Ustawienia:** Można ustawić głośność dźwięków.
- **Wczytaj grę:** Pozwala wczytać grę z zapisanego stanu gry.
- **Autorzy:** Wyświetla autorów aplikacji.
- **Wyjdź z gry:** Bezpieczne zamknięcie aplikacji, zamykające wszystkie aktywne wątki i połączenia sieciowe.

Interfejs Menu Głównego został zaprojektowany z użyciem nowoczesnych komponentów JavaFX, stylizowanych za pomocą CSS

### 3. Lobby (Poczekalnia)

Po wybraniu opcji "Stwórz Grę" lub "Dołącz do gry", użytkownik przenoszony jest do widoku Lobby. Jest to ekran przejściowy służący do połączenia graczy przed rozgrywką.



Rys. 2. Zrzut Ekranu 2 - Widok Lobby przed rozpoczęciem gry

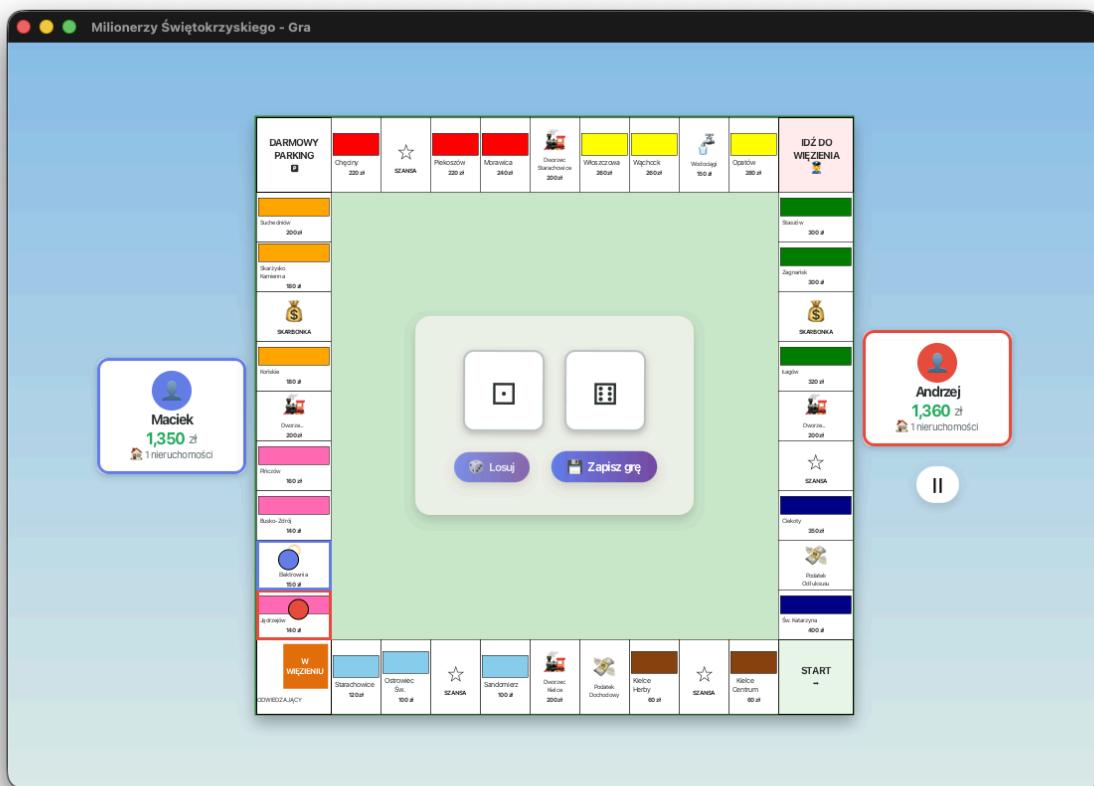
#### Opis elementów ekranu:

- **Lista Graczy:** W centralnej części ekranu wyświetlana jest lista wszystkich osób, które połączły się z serwerem. Każdy wpis zawiera nazwę gracza.
- **Status Sieciowy:** W rogu ekranu widoczna jest informacja o stanie połączenia.
- **Przycisk Start (Tylko Host):** Tylko gracz, który utworzył grę (Host), posiada przycisk umożliwiający rozpoczęcie rozgrywki. Po jego naciśnięciu, do wszystkich klientów wysyłany jest sygnał **START\_GAME**, a aplikacja przechodzi do widoku planszy.
- **Czar Lobby (Opcjonalnie):** Miejsce na krótką komunikację tekstową przed startem gry.

Mechanizm Lobby oparty jest na klasie **LobbyView** oraz **NetworkManager**, które wymieniają komunikaty typu **PLAYER\_LIST** i **CONNECT** w celu aktualizacji listy obecnych na żywo.

## 4. Gra (Główny Widok Rozgrywki)

To najważniejszy widok aplikacji, gdzie toczy się właściwa rozgrywka. Ekran podzielony jest na logiczne sekcje ułatwiające zarządzanie turą.



Rys. 3. Zrzut Ekranu 3 - Główny widok rozgrywki z planszą

### Opis elementów ekranu:

#### 1. Plansza (BoardComponent):

- Centralny element wizualizujący plansze gry z polami.
- Pola reprezentują konkretne miasta (np. Kielce, Ostrowiec, Sandomierz) oraz obiekty specjalne (Dworzec, Elektrownia).
- Po planszy poruszają się pionki graczy (MovePlayerAnimation).
- Pola posiadają kolorowe oznaczenia (obwódki), które zmieniają kolor w momencie wykupienia nieruchomości przez gracza, sygnalizując własność.

#### 2. Panele Graczy (PlayerPanelComponent):

- Umieszczone po bokach planszy.
- Każdy panel wyświetla kluczowe informacje o graczu: Nick, aktualny stan konta, oraz listę posiadanych kart/nieruchomości.
- Panel aktywnego gracza jest wyróżniony graficznie, wskazując czyja jest tura.

#### 3. Panel Sterowania (Środek Planszy):

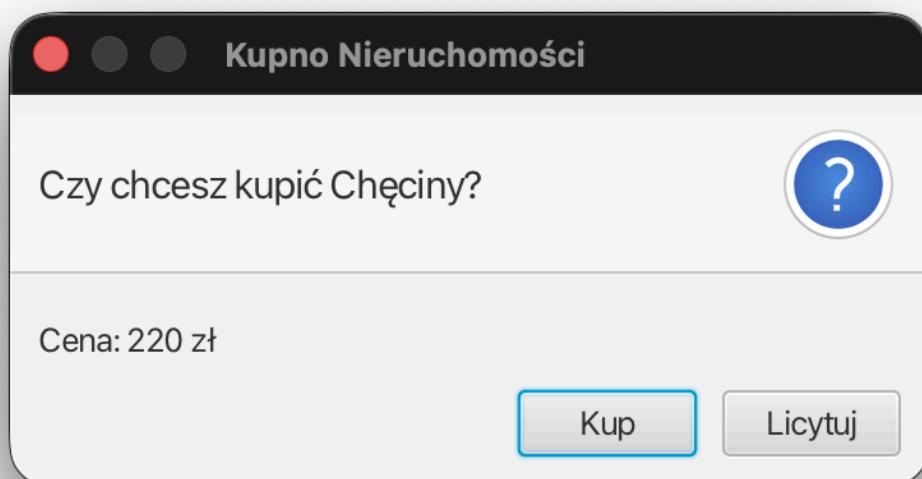
- **Kostki (DiceComponent)**: Wizualizacja rzutu dwiema kośćmi 3D. Animacja rzutu jest synchronizowana sieciowo, aby każdy widział ten sam wynik.
- **Przycisk "Losuj"**: Aktywny tylko na początku tury gracza. Rzut kościami.
- **Przycisk "Zapisz grę"**: Pozwala zapisać grę do pliku. Pokazuje się komunikat, w którym wpisujemy nazwę do pola tekstowego i zatwierdzamy przyciskiem.

#### 4. Logika zdarzeń:

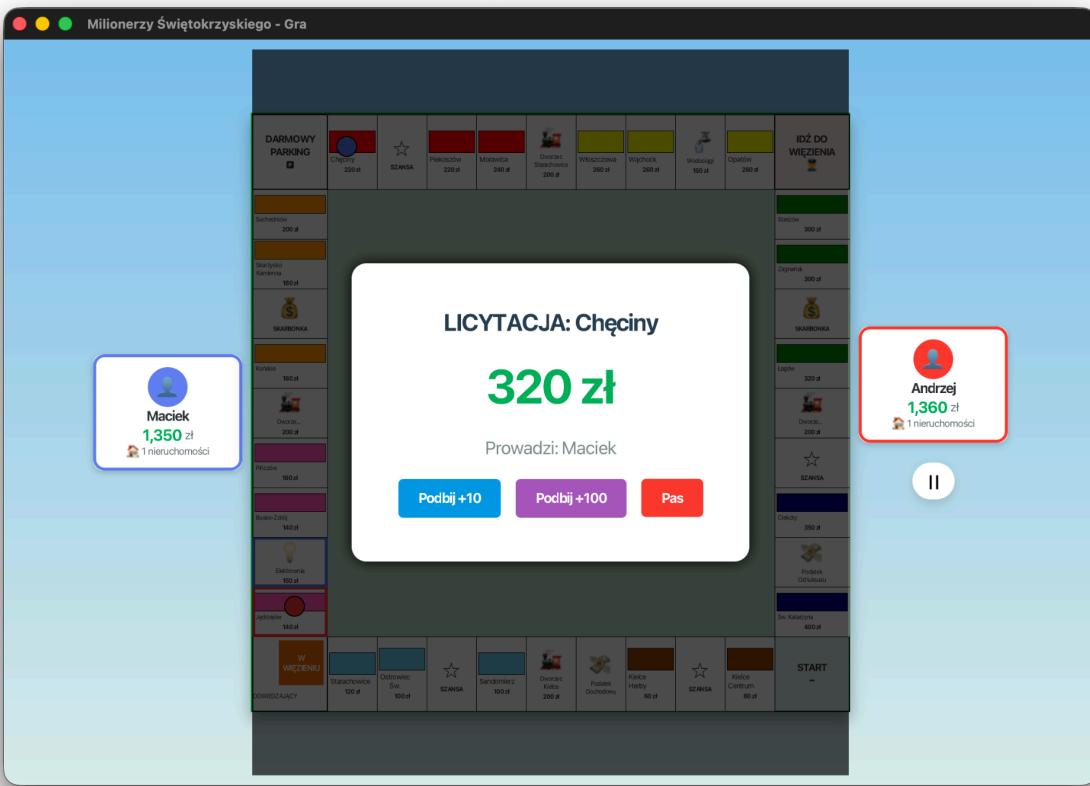
- Gdy gracz stanie na polu należącym do przeciwnika, system automatycznie naliczy czynsz.

## 5. Mechaniki Dodatkowe (Aukcje i Zakupy)

Gra oferuje zaawansowane mechaniki ekonomiczne, w tym system aukcyjny, który uruchamia się, gdy gracz zrezygnuje z zakupu pola, na którym stanął.



Rys. 4. Zrzut Ekranu 4 - Okno dialogowe zakupu nieruchomości



Rys. 5. Zrzut Ekranu 5 - Okno Aukcji

#### Opis mechaniki:

- Okno Zakupu:** Gdy gracz staje na wolnym polu, wyświetlane jest okno dialogowe z informacją o cenie, czynszu i zysku z inwestycji. Gracz może kupić pole lub je odrzucić.
- Widok Aukcji (AuctionComponent):** W przypadku wybrania opcji Licytuj, gra przechodzi w tryb licytacji. Wszyscy gracze widzą specjalny panel aukcyjny z aktualną najwyższą ofertą.
- Licytacja:** Gracze mogą podbijać stawkę. Gdy wszyscy spasują, nieruchomość trafia do osoby, która zaoferowała najczęściej.

## 6. Podsumowanie Techniczne

Projekt został zrealizowany zgodnie ze wzorcem architektonicznym **MVC (Model-View-Controller)**:

- Model:** Logika biznesowa (**GameState**, **Player**, **Board**) jest całkowicie odseparowana od interfejsu. Za logikę odpowiadają dedykowane Menedżery (**MovementManager**, **PropertyManager**).
- View:** Warstwa prezentacji (**GameView**, **BoardComponent**) odpowiada jedynie za wyświetlanie stanu gry i animacje.

- **Network:** Komunikacja sieciowa oparta na przesyłaniu obiektów [GameMessage](#) zapewnia synchronizację stanu gry między wieloma klientami w czasie rzeczywistym.

## 7. Zaimplementowane Funkcjonalności

W ramach projektu zaimplementowano i zweryfikowano poprawne działanie następujących kluczowych mechanizmów:

- **Ruszanie pionków po planszy:** System ruchu jest w pełni zsynchronizowany. Pozycje pionków aktualizują się płynnie u wszystkich podłączonych klientów oraz hosta w czasie rzeczywistym.
- **Kupowanie nieruchomości:** Gracze mają możliwość nabywania wolnych pól. Informacja o zmianie właściciela jest natychmiast rozsyłana do wszystkich uczestników gry.
- **Licytowanie (Aukcje):** Zaimplementowano kompletny system aukcyjny. W przypadku rezygnacji z zakupu, uruchamia się interfejs licytacji dostępny dla wszystkich graczy.
- **Czat w Lobby:** Poczekalnia posiada w pełni funkcjonalny czat tekstowy, umożliwiający komunikację przed rozpoczęciem rozgrywki.
- **Trwałość stanu aplikacji:** Zastosowano mechanizmy serializacji danych ([SaveManager](#)), co pozwala na zachowanie ciągłości stanu gry i jej parametrów.
- **Łączzenie graczy:** Moduł sieciowy ([NetworkManager](#)) obsługuje stabilne nawiązywanie połączeń TCP, poprawnie identyfikując nowych graczy i dodając ich do listy.
- **Synchronizowanie stanu gry:** Wszystkie kluczowe zdarzenia (rzuty kostką, zmiany tur, transakcje finansowe) są synchronizowane pomiędzy wszystkimiinstancjami gry, zapewniając spójność rozgrywki (brak desynchronizacji stanu planszy).