

# Dokumentacja użytkownika komponentu TicTacToeComp

## Wprowadzenie

Komponent TicTacToeComp to gra w kółko i krzyżyk zaimplementowana w bibliotece Swing (Java). Gracz(e) umieszczają symbole „X” lub „O” na planszy 3x3, aby ułożyć trzy symbole w linii (poziomo, pionowo lub po przekątnej). Gra obsługuje trzy tryby: PVP (gracz kontra gracz), EasyAI (łatwy komputer) i HardAI (trudny komputer). Wyniki oraz ustawienia są zapisywane w plikach konfiguracyjnych. Komponent jest konfigurowalny za pomocą klasy TicTacToeConfig.

## Wymagania

- **Środowisko:** Java 8+ z biblioteką Swing.
- **Zależności:** Standardowa biblioteka Javy (pakiety javax.swing, java.awt).
- **Pliki konfiguracyjne:** Plik ticTacToeConfig.txt (zapisywany w katalogu ~/.tictactoe/) dla ustawień gry oraz ticTacToeScores.txt dla wyników.
- **Zasoby:** Ikony cross.png i circle.png w folderze icons/ w zasobach projektu lub w katalogu ~/.tictactoe/icons/. Opcjonalny plik konfiguracyjny ticTacToeConfig.txt w folderze /.

## Instalacja

1. Skompiluj kod źródłowy komponentu (TicTacToeComp.java, TicTacToeConfig.java) w projekcie Java.
2. Upewnij się, że ikony (cross.png, circle.png) znajdują się w folderze icons/ w zasobach projektu lub w katalogu ~/.tictactoe/icons/.
3. Upewnij się, że plik konfiguracyjny ticTacToeConfig.txt (jeśli używany) znajduje się w folderze zasobów projektu lub w katalogu ~/.tictactoe/.
4. Uruchom grę jako aplikację Java.

## Użycie

### Inicjalizacja gry

Utwórz instancję gry:

```
TicTacToeComp game = new TicTacToeComp();
```

lub z callbackiem dla zamknięcia:

```
TicTacToeComp game = new TicTacToeComp(() -> System.out.println("Gra zamknięta"));
```

1. Gra automatycznie inicjalizuje GUI i jest gotowa do użycia po wyświetleniu okna.

## Rozpoczęcie gry

- Kliknij lewym przyciskiem myszy na dowolne puste pole, aby umieścić symbol („X” dla gracza 1, „O” dla gracza 2 lub komputera).
- W trybie EasyAI komputer wykonuje losowe ruchy, w trybie HardAI używa algorytmu minimax.
- Po zakończeniu gry (wygrana lub remis) pojawi się okno dialogowe z opcją kontynuacji lub zamknięcia gry.
- Naciśnij klawisz **Enter** lub kliknij przycisk „Reset”, aby zrestartować grę.

## Sterowanie

- **Lewy przycisk myszy:** Umieszcza symbol na wybranym polu.
- **Klawisz Enter:** Restartuje grę.
- **Przycisk „Reset”:** Restartuje grę z bieżącymi ustawieniami.

## Konfiguracja

Gra korzysta z klasy `TicTacToeConfig` do zarządzania ustawieniami, które można zmieniać programowo lub poprzez plik konfiguracyjny.

### Programowa zmiana ustawień

Pobierz obiekt konfiguracyjny:

*TicTacToeConfig config = game.config; // Pole jest publiczne, ale zaleca się użycie gettera, jeśli zostanie dodany*

1. Dostosuj ustawienia, np.:  
*config.setGameMode("HardAI"); // Tryb trudnego komputera*  
*config.setPlayer1("Anna"); // Nazwa gracza 1*  
*config.setPlayer2("Jan"); // Nazwa gracza 2 (tylko w trybie PVP)*  
*config.setTextColor("#FF0000"); // Czerwony kolor tekstu wyniku*  
*config.setCrossPNG("icons/custom\_cross.png"); // Niestandardowa ikona X*
2. Zapisz ustawienia do pliku:  
*config.saveSettings();*
3. Zrestartuj grę, aby zastosować zmiany:  
*game.restartCurrentGame();*

### Konfiguracja przez plik

Ustawienia są zapisywane w pliku `~/tictactoe/ticTacToeConfig.txt` za pomocą metody:

`config.saveSettings();`

1. Wczytaj ustawienia z pliku podczas inicjalizacji gry (automatyczne przy tworzeniu `TicTacToeComp`):  
*config.loadSettings();*

## Główne metody

- **Konstruktor `TicTacToeComp()`**: Inicjalizuje grę z domyślnymi ustawieniami.
- **Konstruktor `TicTacToeComp(Runnable onExit)`**: Inicjalizuje grę z callbackiem wywoływanym przy zamknięciu.
- **`restartCurrentGame()`**: Restartuje grę z bieżącymi ustawieniami.
- **`closeGame()`**: Zamyka grę, wywołuje callback `onExit` (jeśli ustawiony) i zwalnia zasoby.
- **`saveScores(String player, String gameMode)`**: Zapisuje wynik gracza (zwycięstwo) do pliku `ticTacToeScores.txt`.
- **`resetGame()`**: Czyści planszę, zachowując wynik ogólny.

## Dostępne ustawienia (TicTacToeConfig)

Parametr	Typ	Domyślna wartość	Opis
<code>difficulty</code>	<code>int</code>	0 (EasyAI)	Poziom trudności/tryb gry (0: EasyAI, 1: HardAI, 2: PVP)
<code>player1</code>	<code>String</code>	"Player1"	Nazwa gracza 1 (X)
<code>player2</code>	<code>String</code>	"Player2"	Nazwa gracza 2 (O, tylko w PVP)
<code>gameMode</code>	<code>String</code>	"PVP"	Tryb gry (EasyAI, HardAI, PVP)
<code>circlePNG</code>	<code>String</code>	"icons/circle.png"	Ścieżka do ikony „O”
<code>crossPNG</code>	<code>String</code>	"icons/cross.png"	Ścieżka do ikony „X”
<code>textColor</code>	<code>String</code>	"#000000" (czarny)	Kolor tekstu wyniku (format hex)

## Mechanika gry

- **Plansza:** Siatka 3x3 pól, na których gracze umieszczają symbole „X” lub „O”.
- **Tryby gry:**
  - **PVP (2):** Dwóch graczy na zmianę stawia symbole.
  - **EasyAI (0):** Komputer wykonuje losowe ruchy.
  - **HardAI (1):** Komputer używa algorytmu minimax, przewidując najlepsze ruchy.
- **Zwycięstwo:** Gracz wygrywa, umieszczając trzy symbole w linii (poziomo, pionowo lub po przekątnej). Zwycięska linia jest podświetlana na zielono.
- **Remis:** Gra kończy się remisem, gdy plansza jest pełna i nikt nie wygrał.
- **Wyniki:** Zwycięstwa są zapisywane w pliku `ticTacToeScores.txt` w formacie `gracz, tryb, wynik`. Licznik zwycięstw (`player1Wins`, `player2Wins`) jest wyświetlany w `scoreLabel`.

## Ważne uwagi

- **Fokus klawiatury:** Panel gry (`boardPanel`) musi mieć fokus, aby odbierać zdarzenia klawiatury (np. Enter). Kliknięcie myszą na panel przywraca fokus.
- **Ikony:** Ikony `cross.png` i `circle.png` muszą być dostępne w folderze `icons/` lub `~/tictactoe/icons/`. Brak ikon powoduje wyjątek.
- **Zapis wyników:** Wyniki są zapisywane w pliku `~/tictactoe/ticTacToeScores.txt` w formacie `gracz,difficulty,wynik`.
- **Konfiguracja:** Plik `ticTacToeConfig.txt` jest wczytywany automatycznie przy starcie gry. Jeśli nie istnieje, używane są domyślne wartości.
- **Kolor tekstu:** Kolor tekstu wyniku (`textColor`) musi być w formacie hex (np. `#FF0000`). Nieprawidłowe wartości mogą powodować błędy.

## Przykład pełnego kodu

```
import tictactoe.TicTacToeComp;
public class Main {
    public static void main(String[] args) {
        // Inicjalizacja gry z callbackiem
        TicTacToeComp game = new TicTacToeComp(() -> System.out.println("Gra zamknięta"));
        // Opcjonalna zmiana ustawień
        game.config.setGameMode("HardAI"); // Tryb trudnego komputera
        game.config.setPlayer1("Anna"); // Gracz 1
        game.config.setTextColor("#0000FF"); // Niebieski tekst wyniku
        try {
            game.config.saveSettings();
        } catch (IOException e) {
            System.err.println("Błąd zapisu konfiguracji: " + e.getMessage());
        }
    }
}
```

## Czyszczenie zasobów

Aby zamknąć grę i zwolnić zasoby, wywołaj:

```
game.closeGame();
```

Metoda ta wywołuje callback `onExit` (jeśli ustawiony) i zamyka okno gry.

## Obsługa błędów

- **Brak pliku konfiguracyjnego:** Gra używa domyślnych wartości, jeśli plik `ticTacToeConfig.txt` nie istnieje.
- **Błąd wczytywania/zapisywania:** Wyświetlane są komunikaty błędów w konsoli, a gra kontynuuje działanie z domyślnymi wartościami.
- **Nieprawidłowe wartości w pliku konfiguracyjnym:** Np. nieprawidłowy format liczby lub trybu gry powoduje komunikat błędu, a wartość pozostaje domyślna.
- **Brak ikon:** Brak plików `cross.png` lub `circle.png` powoduje wyjątek `RuntimeException`.

## Rozszerzanie komponentu

- **Dodanie nowych trybów gry:** Rozszerz klasę `TicTacToeConfig`, dodając nowe wartości dla `difficulty` i dostosuj logikę w `handleMove` oraz `computerMove`.
- **Niestandardowe ikony:** Użyj metody `copyImageToIconsDir` do dodawania nowych ikon i zaktualizuj `crossPNG` oraz `circlePNG`.
- **Rozszerzenie planszy:** Zmodyfikuj rozmiar planszy (np. 4x4) w `TicTacToeComp`, dostosowując `buttons` i logikę sprawdzania wygranej.

## Ograniczenia

- Gra obsługuje tylko planszę 3x3 – większe plansze wymagają modyfikacji kodu.
- Brak wsparcia dla pauzy gry lub menu ustawień w GUI.
- Tryb `HardAI` może być wolniejszy na słabszych komputerach z powodu algorytmu `minimax`.