

# Dokumentacja użytkownika komponentu AnimatedBackground

## Wprowadzenie

Komponent AnimatedBackground to dynamiczne tło graficzne oparte na bibliotece JavaFX, które umożliwia wyświetlanie animowanych elementów, takich jak wąż, linie siatki, krzyże, koła oraz efekty eksplozji. Komponent jest konfigurowalny za pomocą klasy AnimationConfig, co pozwala na dostosowanie jego wyglądu i zachowania.

## Wymagania

- **Środowisko:** Java 8+ z JavaFX.
- **Zależności:** Biblioteka JavaFX oraz standardowa biblioteka Javy.
- **Pliki konfiguracyjne:** Plik config.txt (opcjonalny, zapisywany w katalogu ~/.minigames/) dla ustawień niestandardowych.
- **Pliki zasobów:** Plik dźwiękowy (np. /sounds/mainSoundtrack.mp3) dla tła muzycznego, jeśli włączony.

## Instalacja

1. Skompiluj kod źródłowy komponentu (AnimatedBackground.java, AnimationConfig.java, AnimatedSnake.java) w projekcie JavaFX.
2. Upewnij się, że plik dźwiękowy (np. mainSoundtrack.mp3) znajduje się w folderze zasobów (/sounds/).
3. Dodaj komponent do sceny JavaFX w aplikacji.

## Użycie

### Inicjalizacja komponentu

Utwórz instancję komponentu:

```
AnimatedBackground background = new AnimatedBackground();
```

1. Zainicjuj komponent, aby załadować elementy graficzne i animacje:  
`background.init();`
2. Dodaj komponent do sceny JavaFX:  
`Scene scene = new Scene(background, 800, 600);`  
`primaryStage.setScene(scene);`  
`primaryStage.show();`

## Konfiguracja

Komponent korzysta z klasy `AnimationConfig` do zarządzania ustawieniami. Domyślne wartości można zmienić programowo lub poprzez plik konfiguracyjny.

### Programowa zmiana ustawień

Pobierz obiekt konfiguracyjny:

```
AnimationConfig config = background.getConfig();
```

1. Dostosuj ustawienia, np.:

```
config.setBackgroundColor("#000000"); // Czarne tło  
config.setSnakeVisible(true); // Włącz węża  
config.setSnakeColor("#FF0000"); // Czerwony wąż  
config.setMusicEnabled(false); // Wyłącz muzykę
```

2. Zaktualizuj komponent:

```
background.setConfig(config);  
background.updateSettings();
```

3. **Konfiguracja przez plik**

Ustawienia są zapisywane w pliku `~/minigames/config.txt` za pomocą metody:

```
config.saveSettings("config.txt");
```

4. Wczytaj ustawienia z pliku:

```
config.loadSettings("config.txt");  
background.setConfig(config);  
background.updateSettings();
```

## Główne metody

- `init()`: Inicjalizuje komponent, dodając siatkę, węża, koła, krzyże i efekty eksplozji.
- `setConfig(AnimationConfig config)`: Ustawia nowy obiekt konfiguracyjny i aktualizuje komponent.
- `updateSettings()`: Odświeża komponent na podstawie bieżącej konfiguracji.
- `dispose()`: Zatrzymuje animacje, wyłącza odtwarzacz muzyczny i czyści komponent.
- `setAnimatedSnakeColor(String color)`: Zmienia kolor węża.

## Dostępne ustawienia (AnimationConfig)

Parametr	Typ	Domyślna wartość
width	int	800
height	int	600
isMusicEnabled	boolean	true
musicFile	String	"/sounds/mainSoundtrack.mp3"
volMusic	int	50
isCrossVisible	boolean	true
crossSize	int	30
crossColor	String	"#c8c8c8"
isCircleVisible	boolean	true
circleSize	int	20
circleColor	String	"#c8c8c8"
isSnakeVisible	boolean	true
snakeColor	String	"#008000"
snakeLength	int	14
isBackgroundVisible	boolean	true
backgroundColor	String	"#1a1a1a"
isLineVisible	boolean	true
lineColor	String	"#292929"
isExplosionEnabled	boolean	true
minExploDelay	int	1000
maxExploDelay	int	3000

## Animacje

- **Wąż:** Porusza się po zdefiniowanej ścieżce (klasa `AnimatedSnake`). Każdy segment węża to prostokąt 20x20 pikseli. Ścieżka i czas animacji są stałe (7 sekund na pełny obieg).
- **Krzyże i koła:** Pojawiają się losowo z efektem zanikania (3 sekundy). Ich widoczność i kolory można konfigurować.
- **Eksplzje:** Czerwone koła pojawiają się losowo z efektem skalowania i zmiany koloru (czerwony → pomarańczowy). Czas między eksplozjami jest losowy w zakresie `minExploDelay`–`maxExploDelay`.
- **Siatka linii:** Rysuje pionowe i poziome linie co 50 pikseli. Widoczność i kolor linii można zmieniać.

## Ważne uwagi

- **Muzyka:** Jeśli plik muzyczny nie zostanie znaleziony, muzyka jest automatycznie wyłączana (`isMusicEnabled = false`).
- **Eksplzje:** Efekty eksplozji są włączane tylko, gdy `isExplosionEnabled` jest ustawione na `true`.
- **Waż:** Jeśli `isSnakeVisible` jest wyłączone, segmenty węża są niewidoczne, ale nadal istnieją w pamięci.
- **Zasoby:** Upewnij się, że ścieżka do pliku muzycznego jest poprawna, w przeciwnym razie komponent zgłosi wyjątek.

## Przykład pełnego kodu

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.stage.Stage;
import org.example.animatedbackgroundcomp.AnimatedBackground;
import org.example.animatedbackgroundcomp.AnimationConfig;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        AnimatedBackground background = new AnimatedBackground();
        background.init();

        // Opcjonalna zmiana ustawień
        AnimationConfig config = background.getConfig();
        config.setSnakeColor("#FF0000");
        config.setBackgroundColor("#000000");
        background.setConfig(config);
        background.updateSettings();

        Scene scene = new Scene(background, 800, 600);
```

```

        primaryStage.setScene(scene);
        primaryStage.setTitle("Animated Background");
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}

```

## Czyszczenie zasobów

Aby zwolnić zasoby (np. odtwarzacz muzyczny) i zatrzymać animację, wywołaj metodę:

```
background.dispose();
```

## Obsługa błędów

- **Brak pliku muzycznego:** Komponent automatycznie wyłącza muzykę i kontynuuje działanie.
- **Błąd wczytywania pliku konfiguracyjnego:** W przypadku błędów parsowania wartości (np. nieprawidłowy format liczby), wypisywany jest komunikat błędu w konsoli, a wartość pozostaje bez zmian.
- **Nieprawidłowy kolor:** Kolory muszą być podane w formacie hex (np. #FFFFFF). Nieprawidłowe wartości mogą spowodować wyjątek w JavaFX.

## Rozszerzanie komponentu

- **Dodanie nowych animacji:** Można rozszerzyć metodę `init()` lub dodać nowe elementy do Pane w metodzie `addCircleAndCross` lub podobnych.
- **Zmiana ścieżki węża:** Edytuj metodę `init()` w klasie `AnimatedSnake`, aby zmienić obiekt `Path`.
- **Nowe efekty:** Dodaj nowe efekty do metody `playExplosion` lub stwórz własne metody animacji.

## Ograniczenia

- Komponent nie obsługuje dynamicznej zmiany rozmiaru po inicjalizacji (musi być ustawiony w `AnimationConfig` przed wywołaniem `init()`).
- Ścieżka węża jest stała i nie można jej zmieniać bez modyfikacji kodu.
- Brak obsługi interakcji użytkownika (np. kliknięcia myszą na elementy).