

# SYMULACJA ROZPRZESTRZENIANIA SIĘ OGNIA W LESIE

**LIDER: RAFAŁ POPIEL DE CHOSZCZAK 284737, JAKUB OWOC 284724**

REPOZYTORIUM GITHUB: [https://github.com/RavQ7/projekt\\_po](https://github.com/RavQ7/projekt_po)

## SPIS TREŚCI

Opis projektu .....	3
Cel projektu .....	3
Architektura systemu.....	3
Hierarchia klas i dziedziczenie.....	3
Enkapsulacja .....	3
Polimorfizm .....	4
Wzorce projektowe.....	4
Komponenty systemu .....	5
Elementy terenu.....	5
Czynniki środowiskowe.....	5
Klasa Wiatr.....	5
System symulacji .....	6
Klasa Las.....	6
Interfejs użytkownika.....	6
Klasa ModifiedLasFrame.....	6
Klasa LasPanel .....	6
Klasa LegendPanel.....	6
Funkcjonalności symulacji .....	6
Rozprzestrzenianie ognia.....	6
Dynamika pożaru .....	6
Czynniki losowe.....	7
Analiza danych.....	7
Zbieranie statystyk .....	7
Eksport danych.....	7
Zastosowane techniki programistyczne .....	7
JAVADOC.....	7
Diagramy.....	8
Diagram klas.....	8
Diagram obiektów.....	8
Diagram sekwencji.....	9
Diagram Maszyny Stanów .....	10

## OPIS PROJEKTU

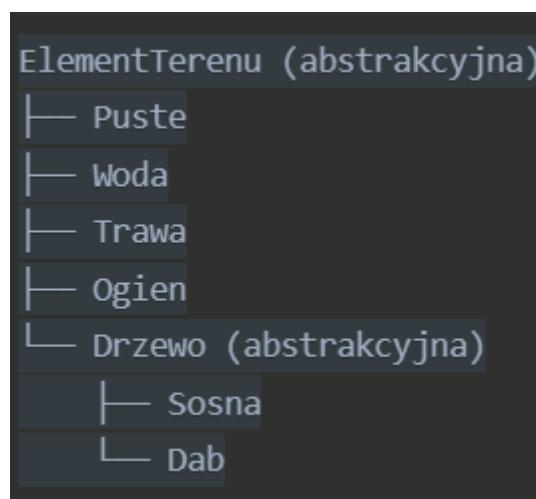
### CEL PROJEKTU

Projekt realizuje symulację dynamiki pożaru w ekosystemie leśnym, reprezentowanym jako dwuwymiarowa siatka różnych elementów terenu. Symulacja umożliwia obserwację procesu rozprzestrzeniania się ognia w zależności od rodzaju terenu, wpływu czynników atmosferycznych oraz interakcji między obiektami.

### ARCHITEKTURA SYSTEMU

#### HIERARCHIA KLAS I DZIEDZICZENIE

Projekt implementuje głęboką hierarchię klas opartą na dziedziczeniu, z abstrakcyjną klasą bazową ElementTerenu jako fundamentem systemu:



Schemat 1

Klasa abstrakcyjna `ElementTerenu` definiuje wspólny interfejs dla wszystkich obiektów występujących na mapie:

- Metody określające palność i możliwość podpalenia (`isPalny`, `canBelgnited`)
- Metodę symulacyjną `nextStep` realizującą logikę danego elementu w kolejnej epoce
- Metodę `stworzKopie` implementującą wzorzec projektowy `Prototype`

#### ENKAPSULACJA

Każda klasa enkapsuluje dane i zachowania właściwe dla danego typu obiektu:

- `Drzewo` zawiera stan drzewa (zdrowe, płonące, spalone), czas palenia i współczynnik palności
- `Wiatr` enkapsuluje kierunek (wektor) i siłę wiatru wraz z metodami modyfikującymi te wartości
- `Las` ukrywa implementację tablicy elementów terenu, udostępniając interfejs do zarządzania symulacją

---

## POLIMORFIZM

Polimorfizm jest intensywnie wykorzystywany w projekcie poprzez:

- Przesłanianie metod abstrakcyjnych z klasy ElementTerenu przez konkretne implementacje
- Używanie referencji typu bazowego do obsługi obiektów różnych typów
- Dynamiczne wiązanie metod podczas wywoływania nextStep i innych metod polimorficznych

---

## WZORCE PROJEKTOWE

W projekcie zastosowano następujące wzorce projektowe:

1. **Prototype** - implementowany przez metodę stwórzKopie w każdej klasie dziedziczącej po ElementTerenu
2. **Model-View-Controller (MVC)** - separacja logiki:
  - Model: klasy Las, ElementTerenu i pochodne
  - Widok: klasy LasPanel, LegendPanel
  - Kontroler: klasa ModifiedLasFrame

## KOMPONENTY SYSTEMU

### ELEMENTY TERENU

#### KLASA ELEMENT TERENU

Abstrakcyjna klasa bazowa definiująca wspólne właściwości wszystkich elementów na mapie:

- Symbol do reprezentacji tekstowej
- Metody abstrakcyjne określające zachowanie podczas symulacji

#### KLASA DRZEWO

Rozszerza ElementTerenu i implementuje wspólną logikę dla wszystkich gatunków drzew:

- Definiuje stany drzewa jako enumerację StanDrzewa (ZDROWE, PLONACE, SPALONE)
- Implementuje logikę rozprzestrzeniania ognia, uwzględniając wpływ wiatru
- Parametryzuje czas palenia i współczynnik palności dla różnych gatunków

#### KLASY SOSNA I DĄB

Konkretne implementacje drzew z własnymi parametrami:

- Sosna: średni czas palenia (5 kroków), wysoka palność (0.4)
- Dab: długi czas palenia (8 kroków), niższa palność (0.25)

#### KLASA TRAWA

Element terenu o niskiej odporności na ogień i krótkim czasie palenia (1 krok).

#### KLASA WODA

Niepalny element będący naturalną barierą dla ognia.

#### KLASA PUSTE

Reprezentuje puste pole, które nie może być podpalone.

## CZYNNIKI ŚRODOWISKOWE

### KLASA WIATR

Modeluje wpływ wiatru na rozprzestrzenianie się ognia:

- Kierunek wyrażony jako wektor dwuwymiarowy (dr, dc)
- Siła w skali 0-5
- Zwiększa prawdopodobieństwo zapłonu w kierunku zgodnym z wiatrem

## SYSTEM SYMULACJI

---

### KLASA LAS

Centralna klasa odpowiedzialna za przechowywanie i zarządzanie stanem symulacji:

- Dwuwymiarowa tablica elementów terenu
- Metody do inicjalizacji, symulacji kolejnych kroków i zbierania statystyk
- Eksport danych do pliku CSV dla późniejszej analizy
- Losowa inicjalizacja terenu z różnymi typami elementów

## INTERFEJS UŻYTKOWNIKA

---

### KLASA MODIFIEDLASFRAME

Główne okno aplikacji zawierające:

- Panel wizualizacji lasu
- Panel legendy kolorów
- Przyciski kontrolne do sterowania symulacją
- Etykiety wyświetlające statystyki i informacje o wietrze

### KLASA LASPANEL

Komponent wizualizujący las jako siatkę kolorowych komórek, realizujący wzorzec MVC jako widok.

### KLASA LEGENDPANEL

Panel wyświetlający legendę kolorów używanych w symulacji.

## FUNKCJONALNOŚCI SYMULACJI

---

### ROZPRZESTRZENIANIE OGNI

- Ogień rozprzestrzenia się z określonym prawdopodobieństwem zależnym od typu elementu
- Wpływ wiatru zwiększa prawdopodobieństwo zapłonu w kierunku zgodnym z kierunkiem wiatru
- Każdy element o stanie PLONACE może podpalić sąsiednie palne elementy

### DYNAMIKA POŻARU

- Drzewa przechodzą przez trzy stany: zdrowe → płonące → spalone
- Każdy gatunek drzewa ma różny czas palenia i współczynnik palności
- Trawa po spaleniu zamienia się w puste pole

- Woda stanowi barierę niemożliwą do przekroczenia przez ogień

---

#### CZYNNIKI LOSOWE

- Losowe rozmieszczenie elementów terenu przy inicjalizacji
- Losowy wybór punktu początkowego pożaru
- Losowy kierunek i siła wiatru w każdej epoce symulacji

#### ANALIZA DANYCH

---

#### ZBIERANIE STATYSTYK

- Zliczanie liczby drzew w poszczególnych stanach
- Śledzenie procentu spalonego lasu
- Monitorowanie długości trwania pożaru

---

#### EKSPORT DANYCH

- Zapisywanie stanu symulacji do pliku CSV w formacie:

*Epoka,Zdrowe,Plonace,Spalone*

- Możliwość dalszej analizy danych przy użyciu zewnętrznych narzędzi

#### ZASTOSOWANE TECHNIKI PROGRAMISTYCZNE

- **Dziedziczenie** dla tworzenia hierarchii klas
- **Enkapsulacja** danych i zachowań w klasach
- **Polimorfizm** do dynamicznego wyboru implementacji metod
- **Abstrakcja** dla definiowania wspólnych interfejsów
- **Wzorce projektowe** (Prototype, MVC)
- **Programowanie zdarzeniowe** w interfejsie graficznym
- **Obsługa wyjątków** dla zwiększenia niezawodności aplikacji
- **Wątki** do separacji logiki symulacji od interfejsu użytkownika

#### JAVADOC

W folderze JAVADOC znajduje się wygenerowana automatycznie dokumentacja do naszego projektu przez funkcję generate Javadoc w programie IntelliJ.

## DIAGRAMY

Wszystkie diagramy również, załączamy w folderze diagramy w celu lepszej jakości zdjęć, gdzie można przybliżyć poszczególne diagramy i nie zachodzi żadna kompresja ani utrata jakości.

## DIAGRAM KLAS

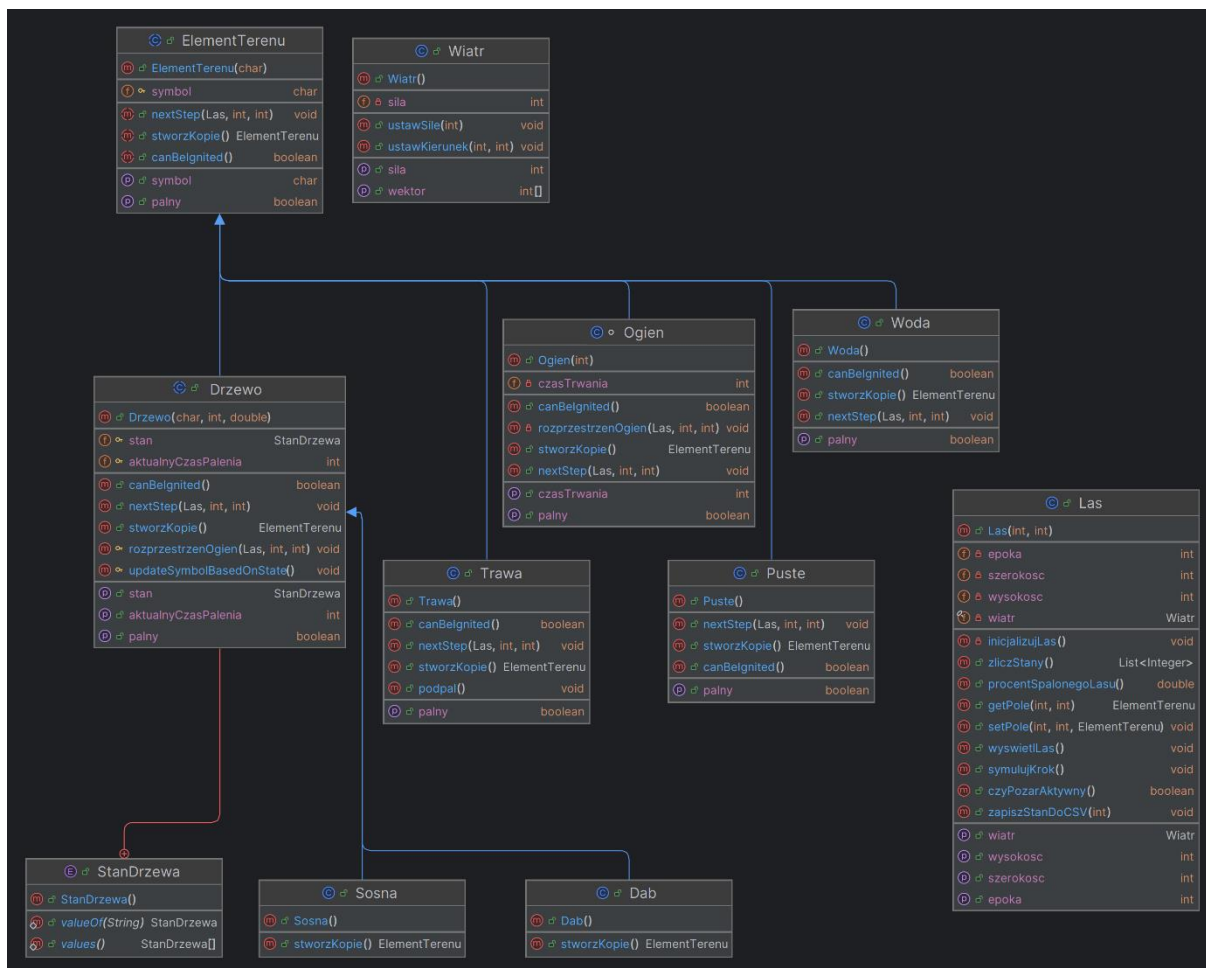


Diagram 1

## DIAGRAM OBIEKTÓW

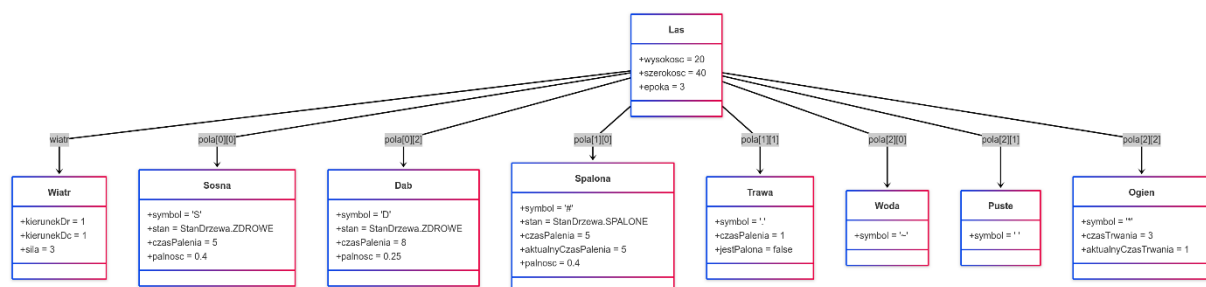


Diagram 2



## DIAGRAM SEKWENCJI

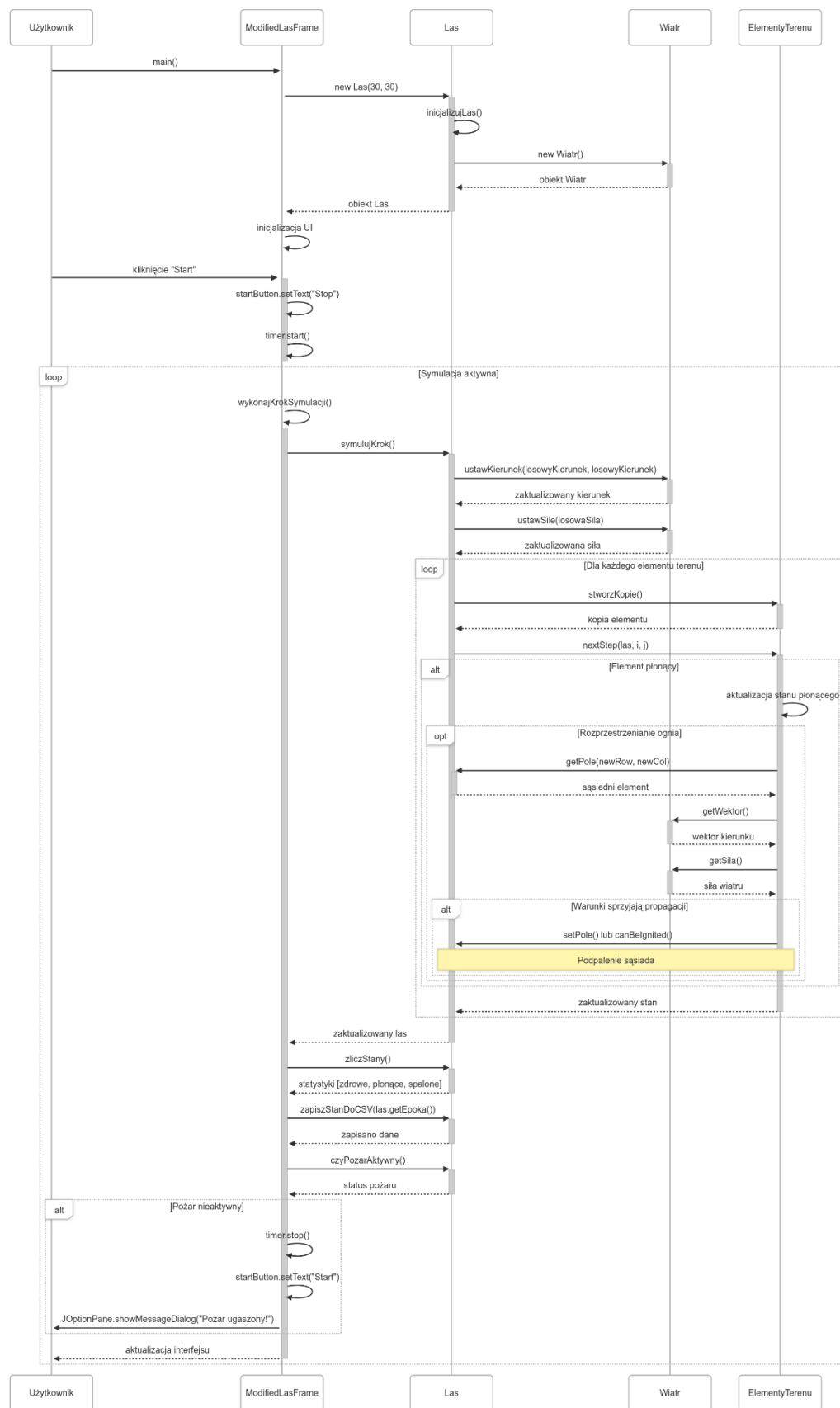


Diagram 3

## DIAGRAM MASZyny STANÓW

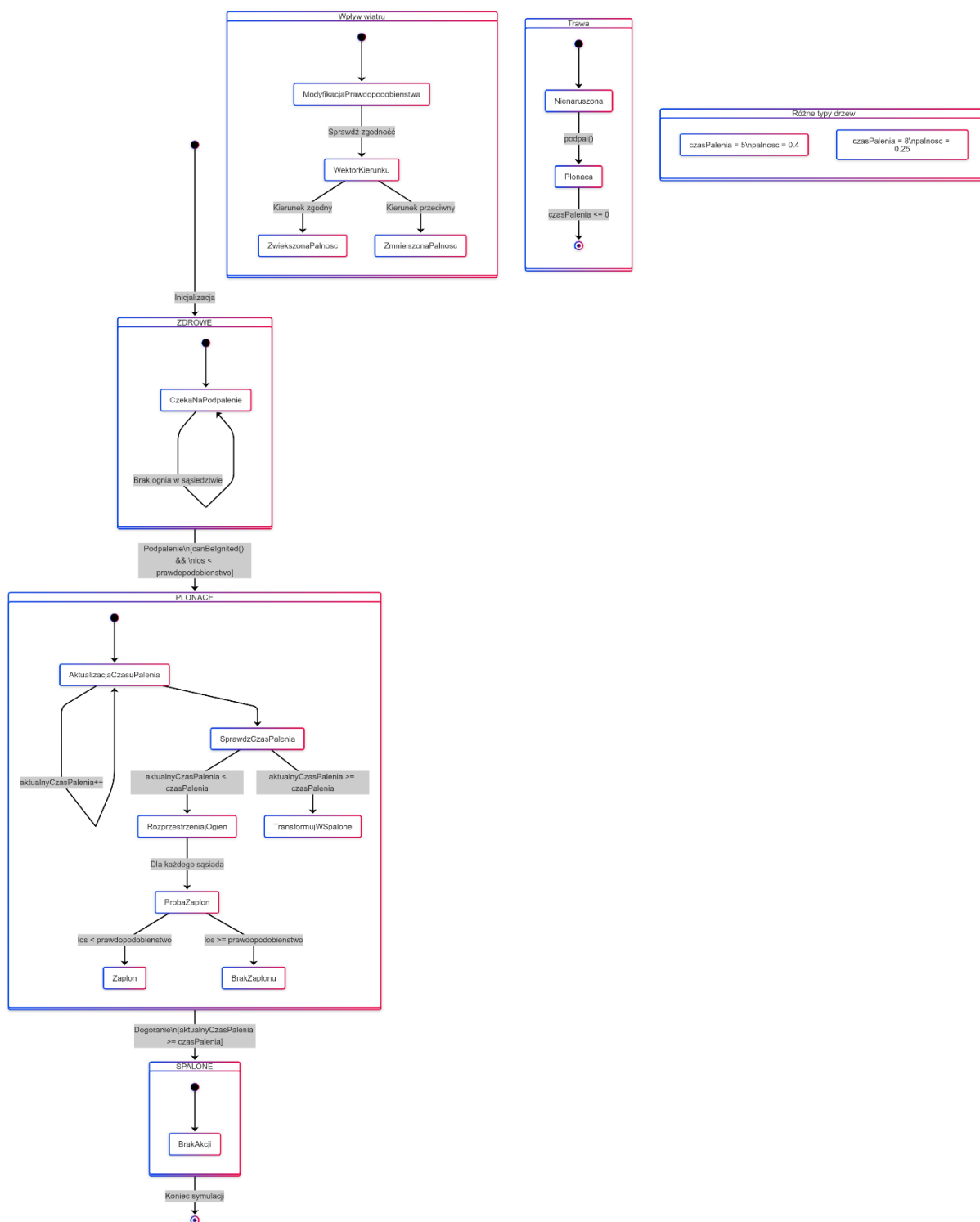


Diagram 4