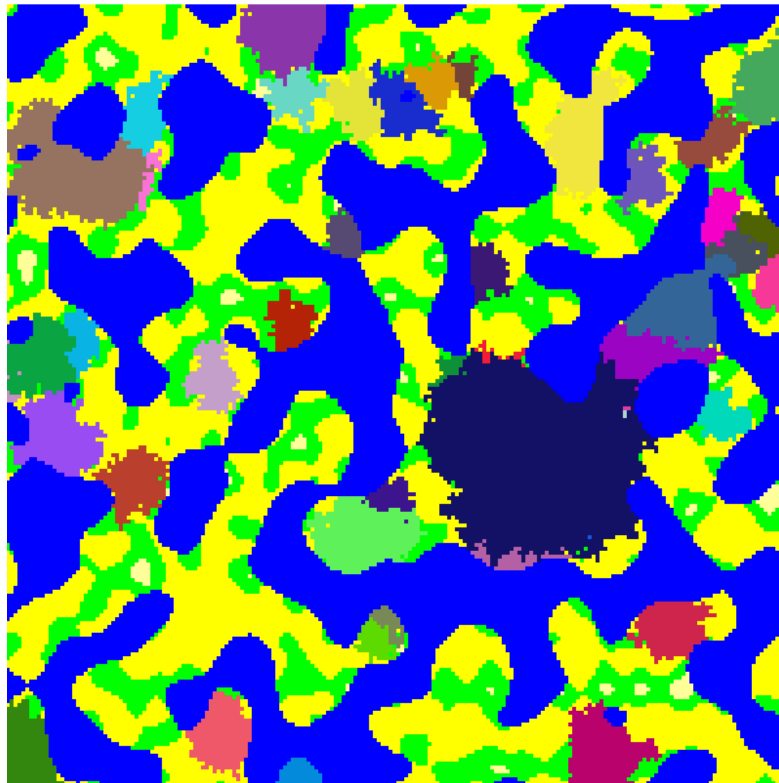


# Symulator plemion

Błażej Balczun

Jakub Michałowicz

Projekt Programowanie Obiektowe



## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Opis projektu . . . . .	2
1.2	Realizacja . . . . .	3
<b>2</b>	<b>Analiza czasowniko rzeczownikowa</b>	<b>5</b>
<b>3</b>	<b>Karty CRC</b>	<b>6</b>
<b>4</b>	<b>Diagramy</b>	<b>7</b>
	Diagram klas . . . . .	7
4.1	Diagram użycia . . . . .	9
4.2	Diagram sekwencji . . . . .	10
4.3	Diagram aktywności . . . . .	11
4.4	Diagram maszyny stanów . . . . .	12
<b>5</b>	<b>Dokumentacja sphinx</b>	<b>12</b>

## 1 Wstęp

Symulator ma na celu symulowanie rozwoju plemion na losowo wygenerowanej mapie. Przy tworzeniu pod uwagę brano skalaność i łatwość konfiguracji systemu.

### 1.1 Opis projektu

Celem projektu jest zasymulowanie zachowań plemion walczących o przetrwanie w epoce kamienia. Na symulację będą składały się plemiona koczownicze i osadnicze, o usposobieniu agresywnym lub pasywnym. Plemiona będą rozwijały się poprzez atakowanie innych plemion i asymilację wędrowców. Każde plemię będzie posiadało własne terytorium w obrębie którego będzie atakować inne plemiona. Na początku symulacji stworzymy kilka-  
naście plemion koczowniczych z odpowiednim usposobieniem. Plemiona te przekształcą

się w plemiona osadnicze lub zginą, a następnie będą rozrastać się do momentu napotkania innego plemienia osadniczego które zaatakują. Na końcu symulacji powinniśmy zostać z modelem stale rozrastającej się wioski osadniczej.

## 1.2 Realizacja

Każdy typ plemienia to obiekt, rozdzielone na dwie klasy, dodatkowo po tablicy chodzą wędrowcy, którzy przyłączają się do pierwszego napotkanego plemienia, lub zginą po upływie kilku "tur".

Symulację przeprowadzamy zaczynając od wygenerowania tablicy, umieszczenia na tablicy plemion i wędrowców oraz określenia ich liczebności i nazwy. Następnie każde plemię agresywne będzie miało szansę na zaatakowanie innego plemienia z którym wejdzie w kontakt na tablicy. Plemię osadnicze atakuje inne plemię, które wejdzie w obręb terytorium tego plemienia.

Jeśli istnieje inne plemię osadnicze w obrębie tego samego terytorium, to plemiona atakują siebie nawzajem.

Plemiona osadnicze pasywne nie przemieszczają się wcale, plemiona koczownicze agresywne stale się przemieszczają oraz mają licznik "tur" po upływie których giną, lub przekształcają się w plemię osadnicze.

Z każdą turą plemię ma szansę na powiększenie terytorium lub liczby członków. Walka pomiędzy plemionami to zwykle porównywanie liczebności plemion, oraz rzut kością sześciocienną dającą plemionom od 1-6 siły ataku, liczniejsze plemię wygrywa i "zabija" jednego członka przegranego plemienia.



## 2 Analiza czasowniko rzeczownikowa

Obiekt	Czasownik	Rzeczownik
Plemię	<ul style="list-style-type: none"> <li>– Zarządzanie</li> <li>– Walka</li> </ul>	<ul style="list-style-type: none"> <li>– Pozycja startowa</li> <li>– Okupowane terytoria</li> <li>– Modyfikatory plemienia</li> </ul>
Mapa	<ul style="list-style-type: none"> <li>– Operacje logiczne na mapie</li> <li>– Pośredniczenie w zmianach własności</li> </ul>	<ul style="list-style-type: none"> <li>– Lista plemion</li> <li>– Lista terenów</li> </ul>
Generator Mapy	<ul style="list-style-type: none"> <li>– Generowanie mapy</li> <li>– Tworzenie plemion</li> <li>– Losowanie modyfikatorów</li> </ul>	<ul style="list-style-type: none"> <li>– Lista plemion</li> <li>– Mapa</li> <li>– Dostępne modyfikatory</li> </ul>
Konfiguracja	<ul style="list-style-type: none"> <li>– Wczytywanie różnych typów konfiguracji</li> <li>– Jednolity interfejs dostępu</li> </ul>	<ul style="list-style-type: none"> <li>– Konfiguracja w formie słownika</li> </ul>
Symulator	<ul style="list-style-type: none"> <li>– Symulowanie tur</li> </ul>	<ul style="list-style-type: none"> <li>– Konfiguracja</li> </ul>

### 3 Karty CRC

Plemię	
Odpowiedzialności	Współtowarzysze
<ul style="list-style-type: none"> <li>– Zarządzanie własnymi terytoriami</li> <li>– Przeprowadzanie wojen</li> <li>– Produkcja zasobów</li> </ul>	<ul style="list-style-type: none"> <li>– Terytorium</li> <li>– Mapa</li> <li>– PomocnikPlemiona</li> </ul>

Mapa	
Odpowiedzialności	Współtowarzysze
<ul style="list-style-type: none"> <li>– Udostępnianie mapy dla plemion (przekazywanie,zdobywanie terytoriów)</li> <li>– Udostępnianie metod do analizy mapy</li> </ul>	<ul style="list-style-type: none"> <li>– Terytorium</li> </ul>

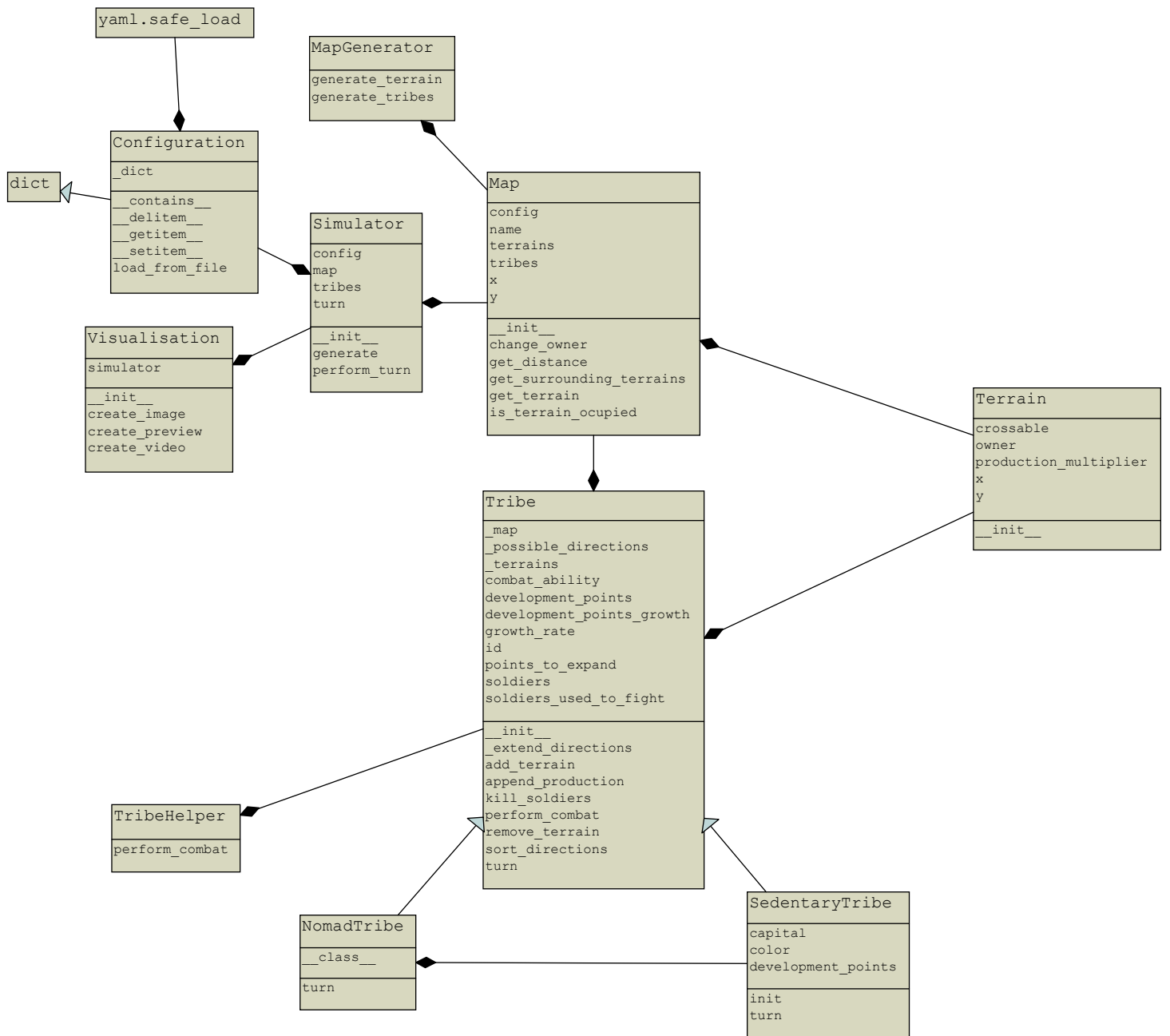
Terytorium	
Odpowiedzialności	Współtowarzysze
<ul style="list-style-type: none"> <li>– Dodawanie produkcji do plemiona</li> </ul>	<ul style="list-style-type: none"> <li>– Plemię</li> </ul>

Generator Mapy	
Odpowiedzialności	Współtowarzysze
<ul style="list-style-type: none"> <li>– Generowanie mapy – tworzenie terytoriów</li> <li>– Generowanie plemion</li> </ul>	<ul style="list-style-type: none"> <li>– Mapa</li> <li>– Konfiguracja</li> </ul>

Symulator	
Odpowiedzialności	Współtowarzysze
<ul style="list-style-type: none"> <li>– Symulowanie tur</li> <li>–</li> </ul>	<ul style="list-style-type: none"> <li>– Mapa</li> <li>– Konfiguracja</li> </ul>

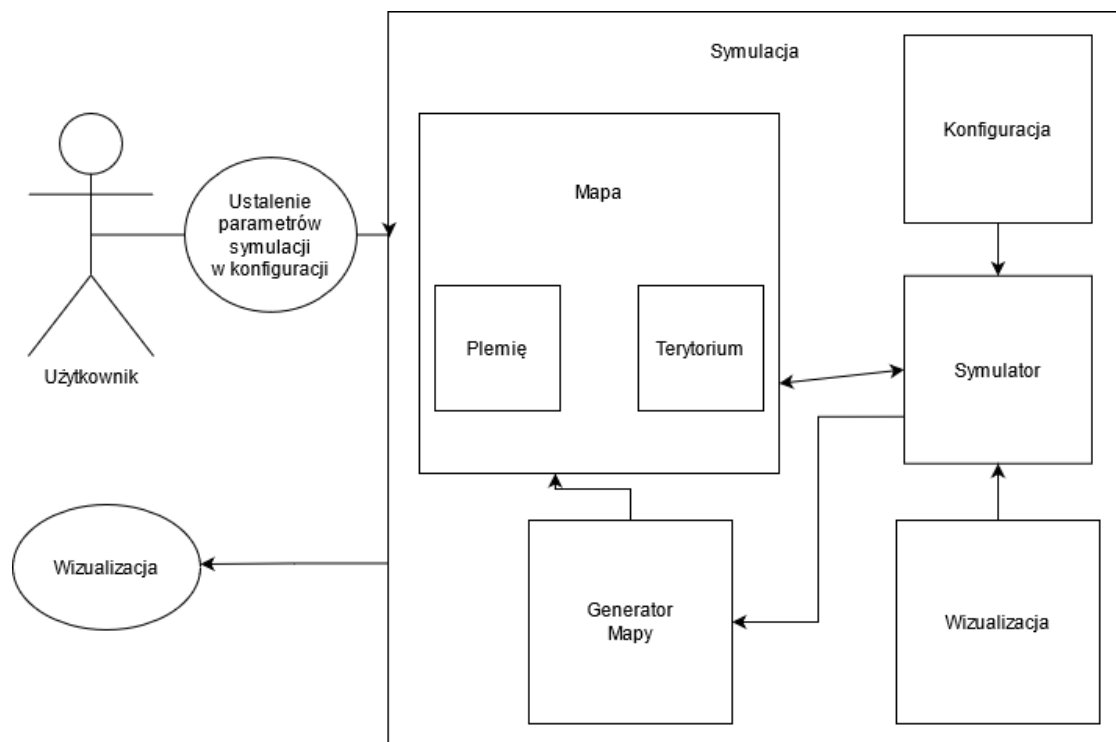
Wizualizator	
Odpowiedzialności	Współtowarzysze
<ul style="list-style-type: none"> <li>– Tworzenie wizualizacji stanu symulacji</li> <li>– Tworzenie filmów wideo</li> </ul>	<ul style="list-style-type: none"> <li>– Symulator</li> </ul>

## 4 Diagramy

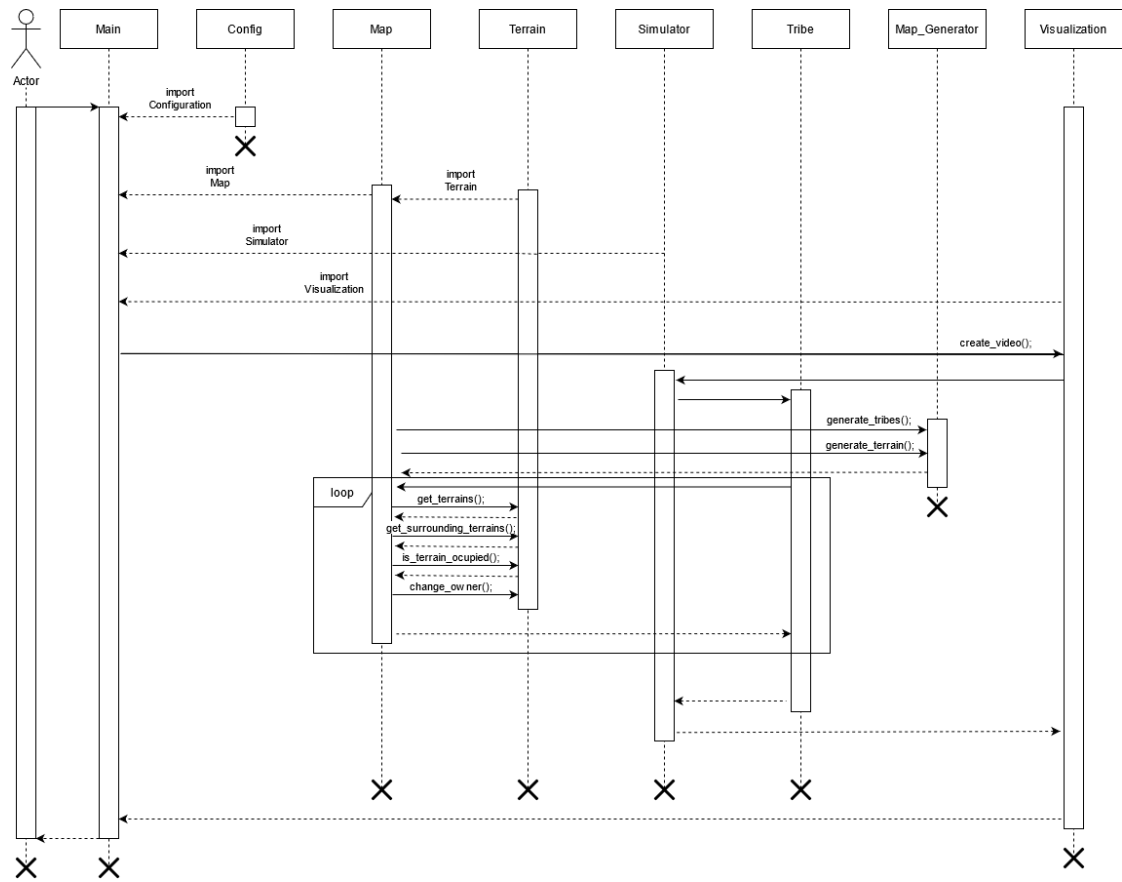




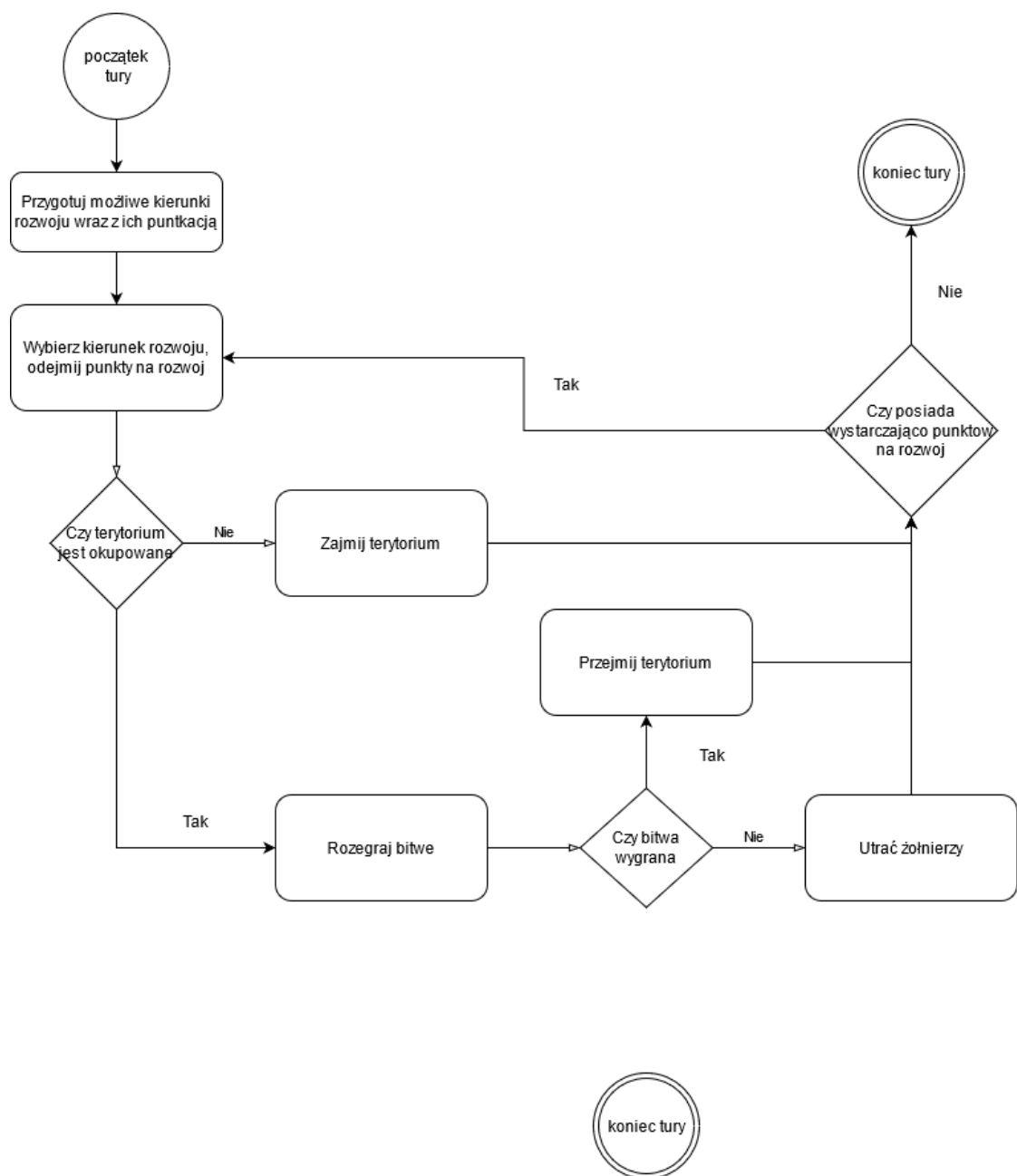
## 4.1 Diagram użycia



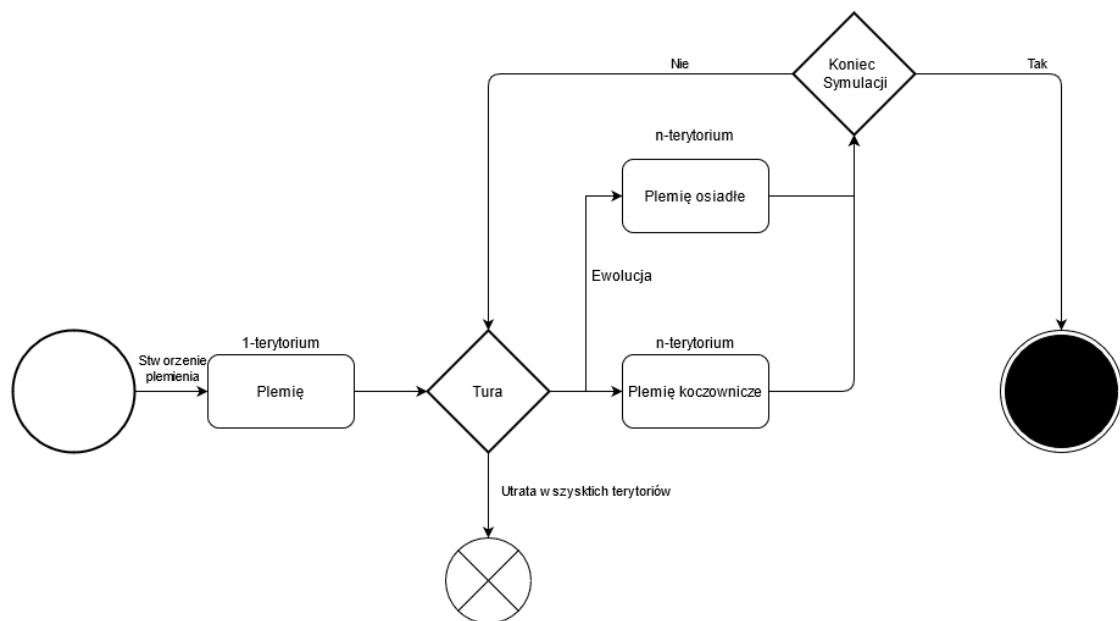
## 4.2 Diagram sekwencji



### 4.3 Diagram aktywności



#### 4.4 Diagram maszyny stanów



#### 5 Dokumentacja sphinx

---

**Plemiona**

**k**

**Jun 17, 2021**



# CONTENTS

1	config	1
2	map	3
3	map_generator	5
4	simulator	7
5	terrain	9
6	tribe	11
7	visualization	13
	Python Module Index	15
	Index	17





---

<i>Configuration</i>	Klasa przygotowana pod możliwość rozszerzenia do innego formatu np Json
----------------------	---

---

## 1.1 Plemiona.config.Configuration

### **class** Plemiona.config.Configuration

Klasa przygotowana pod możliwość rozszerzenia do innego formatu np Json

**`__init__`**(\*args, \*\*kwargs)

Initialize self. See help(type(self)) for accurate signature.

#### **Methods**

---

<b><code>__init__</code></b> (*args, **kwargs)	Initialize self.
<b><code>clear</code></b> ()	
<b><code>copy</code></b> ()	
<b><code>fromkeys</code></b> ([value])	Create a new dictionary with keys from iterable and values set to value.
<b><code>get</code></b> (key[, default])	Return the value for key if key is in the dictionary, else default.
<b><code>items</code></b> ()	
<b><code>keys</code></b> ()	
<b><code>load_from_file</code></b> (file)	
<b><code>pop</code></b> (k[,d])	If key is not found, default is returned if given, otherwise KeyError is raised
<b><code>popitem</code></b> ()	Remove and return a (key, value) pair as a 2-tuple.
<b><code>setdefault</code></b> (key[, default])	Insert key with a value of default if key is not in the dictionary.

---

continues on next page

Table 2 – continued from previous page

<code>update([E, ]**F)</code>	If E is present and has a <code>.keys()</code> method, then does: for k in E: $D[k] = E[k]$ If E is present and lacks a <code>.keys()</code> method, then does: for k, v in E: $D[k] = v$ In either case, this is followed by: for k in F: $D[k] = F[k]$
<code>values()</code>	

---

---

`Map(name, config)`


---

## 2.1 Plemiona.map.Map

**class** `Plemiona.map.Map`(*name*, *config*)

`__init__`(*name*, *config*)

### Parameters

- **name** (*str*) – The name of the map (Pangea)
- **config** (*Config*) – Instance of configuration class

### Methods

---

`__init__`(*name*, *config*)

**param name** The name of the map (Pangea)

---

`change_owner`(*x*, *y*, *owner*)

Method used to change ownership between tribes  
:param *x*: Horizontal position of terrain :type *x*: int  
:param *y*: Horizontal position of terrain :type *y*: int  
:param *owner*: Tribe we want to transfer ownership to :type *owner*: Tribe

---

`get_surrounding_terrains`(*terrain*)

Method used to get surrounding terrains by terrain

---

`get_terrain`(*x*, *y*)

Method used to get terrain by coordinates

---

`is_terrain_occupied`(*x*, *y*)

Method used to check if terrain is occupied.

---



## MAP\_GENERATOR

---

*MapGenerator()*

---

### 3.1 Plemiona.map\_generator.MapGenerator

**class** Plemiona.map\_generator.**MapGenerator**

**`__init__()`**  
Initialize self. See help(type(self)) for accurate signature.

#### Methods

<code><i>__init__()</i></code>	Initialize self.
<code>generate_terrain(map)</code>	Generating terrain with params given in configuration
<code>generate_tribes(map)</code>	Generates tribes according to configuration given in map



## SIMULATOR

---

*Simulator*(*config*)

---

### 4.1 Plemiona.simulator.Simulator

**class** Plemiona.simulator.Simulator(*config*)

**\_\_init\_\_**(*config*)  
Initialize self. See help(type(self)) for accurate signature.

#### Methods

---

<b>__init__</b> ( <i>config</i> )	Initialize self.
-----------------------------------	------------------

---

**generate**()

---

**perform\_turn**([*turns*])

---





TERRAIN

---

*Terrain*(dictionary)

---

5.1 Plemiona.terrain.Terrain

**class** Plemiona.terrain.**Terrain**(*dictionary*)

**\_\_init\_\_**(*dictionary*)  
        Initialize self. See help(type(self)) for accurate signature.

Methods

---

<i>__init__</i> (dictionary)	Initialize self.
------------------------------	------------------

---

Attributes

---

production\_multiplier

---



---

*NomadTribe*(map, dictionary)

---

*SedentaryTribe*(map, dictionary)

---

*Tribe*(map, dictionary)

---

## 6.1 Plemiona.tribe.NomadTribe

**class** Plemiona.tribe.NomadTribe(*map*: Plemiona.map.Map, *dictionary*)

**\_\_init\_\_**(*map*: Plemiona.map.Map, *dictionary*)  
Initialize self. See help(type(self)) for accurate signature.

### Methods

<code>__init__</code> (map, dictionary)	Initialize self.
<code>add_terrain</code> (terrain)	Function used to add terrain to tribe
<code>append_production</code> ()	
<code>kill_soldiers</code> (n)	Function used to decrease number of soldier available to tribe.
<code>perform_combat</code> (enemy)	Function used to perform combat between two tribes.
<code>remove_terrain</code> (terrain)	Remove terrain from tribe terrains and decrease development points growth.
<code>turn</code> ()	Abstract method used to process turn

## 6.2 Plemiona.tribe.SedentaryTribe

**class** Plemiona.tribe.SedentaryTribe(*map*: Plemiona.map.Map, *dictionary*)

**\_\_init\_\_**(*map*: Plemiona.map.Map, *dictionary*)  
Initialize self. See help(type(self)) for accurate signature.

### Methods

<code>__init__(map, dictionary)</code>	Initialize self.
<code>add_terrain(terrain)</code>	Function used to add terrain to tribe
<code>append_production()</code>	
<code>init()</code>	
<code>kill_soldiers(n)</code>	Function used to decrease number of soldier available to tribe.
<code>perform_combat(enemy)</code>	Function used to perform combat between two tribes.
<code>remove_terrain(terrain)</code>	Remove terrain from tribe terrains and decrease development points growth.
<code>turn()</code>	Abstract method used to process turn

## 6.3 Plemiona.tribe.Tribe

**class** Plemiona.tribe.Tribe(*map*: Plemiona.map.Map, *dictionary*)

**\_\_init\_\_**(*map*: Plemiona.map.Map, *dictionary*)  
Initialize self. See help(type(self)) for accurate signature.

### Methods

<code>__init__(map, dictionary)</code>	Initialize self.
<code>add_terrain(terrain)</code>	Function used to add terrain to tribe
<code>append_production()</code>	
<code>kill_soldiers(n)</code>	Function used to decrease number of soldier available to tribe.
<code>perform_combat(enemy)</code>	Function used to perform combat between two tribes.
<code>remove_terrain(terrain)</code>	Remove terrain from tribe terrains and decrease development points growth.
<code>turn()</code>	Abstract method used to process turn

## VISUALIZATION

---

*Visualisation*(simulator)

---

### 7.1 Plemiona.visualization.Visualisation

**class** Plemiona.visualization.**Visualisation**(simulator)

**\_\_init\_\_**(simulator) → None  
Initialize self. See help(type(self)) for accurate signature.

**Methods**

<u><b>__init__</b></u> (simulator)	Initialize self.
<b>create_image</b> ()	Function used to generate image of current simulation status (with tribes and free terrains)
<b>create_preview</b> ()	

---



## PYTHON MODULE INDEX

### p

Plemiona, ??  
Plemiona.config, 1  
Plemiona.map, 3  
Plemiona.map\_generator, 5  
Plemiona.simulator, 7  
Plemiona.terrain, 9  
Plemiona.tribe, 11  
Plemiona.visualization, 13





## Symbols

[\\_\\_init\\_\\_\(\) \(Plemiona.config.Configuration method\), 1](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.map.Map method\), 3](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.map\\_generator.MapGenerator method\), 5](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.simulator.Simulator method\), 7](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.terrain.Terrain method\), 9](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.tribe.NomadTribe method\), 11](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.tribe.SedentaryTribe method\), 12](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.tribe.Tribe method\), 12](#)  
[\\_\\_init\\_\\_\(\) \(Plemiona.visualization.Visualisation method\), 13](#)

## C

[Configuration \(class in Plemiona.config\), 1](#)

## M

[Map \(class in Plemiona.map\), 3](#)  
[MapGenerator \(class in Plemiona.map\\_generator\), 5](#)  
[module](#)  
     [Plemiona, 1](#)  
     [Plemiona.config, 1](#)  
     [Plemiona.map, 3](#)  
     [Plemiona.map\\_generator, 5](#)  
     [Plemiona.simulator, 7](#)  
     [Plemiona.terrain, 9](#)  
     [Plemiona.tribe, 11](#)  
     [Plemiona.visualization, 13](#)

## N

[NomadTribe \(class in Plemiona.tribe\), 11](#)

## P

[Plemiona](#)  
     [module, 1](#)  
[Plemiona.config](#)  
     [module, 1](#)  
[Plemiona.map](#)  
     [module, 3](#)  
[Plemiona.map\\_generator](#)

[module, 5](#)  
[Plemiona.simulator](#)  
     [module, 7](#)  
[Plemiona.terrain](#)  
     [module, 9](#)  
[Plemiona.tribe](#)  
     [module, 11](#)  
[Plemiona.visualization](#)  
     [module, 13](#)

## S

[SedentaryTribe \(class in Plemiona.tribe\), 12](#)  
[Simulator \(class in Plemiona.simulator\), 7](#)

## T

[Terrain \(class in Plemiona.terrain\), 9](#)  
[Tribe \(class in Plemiona.tribe\), 12](#)

## V

[Visualisation \(class in Plemiona.visualization\), 13](#)