

**Uniwersytet Warszawski**  
Wydział Matematyki, Informatyki i Mechaniki

**Nieistniejący Nichilista**

Nr albumu: 123456

# **Implementacja systemu AI-Arena**

Praca licencjacka  
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem  
**dra. Roberta Dabrowskiego**  
Wydział Matematyki Informatyki i Mechaniki

Maj 2012

## **Oświadczenie kierującego pracą**

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

## **Oświadczenie autora (autorów) pracy**

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

### **Streszczenie**

W pracy przedstawiono implementację systemu AI-Arena, służącego do przeprowadzania turniejów programów walczących.

### **Słowa kluczowe**

programy walczące, arena, sztuczna inteligencja

### **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

11.3 Informatyka

### **Klasyfikacja tematyczna**

D. Software  
D.0. General

### **Tytuł pracy w języku angielskim**

Implementation of AI-Arena system



# Spis treści

<b>Wprowadzenie</b>	5
<b>1. Podstawowe pojęcia</b>	7
1.1. Definicje	7
<b>2. Zastosowania w nauce i biznesie internetowym</b>	9
<b>3. Podobne platformy</b>	11
3.1. Top Coder	11
3.2. Ai Challenge	11
3.3. SPOJ	11
<b>4. Use cases</b>	13
<b>5. Metodyka Scrum</b>	17
<b>6. Architektura systemu</b>	19
6.1. Nadzorca	19
6.2. Scheduler	19
6.3. Baza danych	19
6.4. API użytkownika	19
<b>7. Dokumentacja użytkowa i opis implementacji</b>	21
<b>8. Podsumowanie</b>	23
<b>9. Podział prac</b>	25
<b>10. Spis płyty</b>	27
<b>A. Przykładowa gra</b>	29
<b>B. Przykładowe programy</b>	31
<b>C. Przebieg przykładowego turnieju</b>	33
<b>Bibliografia</b>	35



# Wprowadzenie

System Ai-Arena służy do przeprowadzania rozgrywek i turniejów różnych gier pomiędzy programami komputerowymi. System ma w zamierzeniu twórców służyć osobom zainteresowanym sztuczną inteligencją do sprawdzenia swoich umiejętności, lub jako pomoc przy badaniach nad sztuczną inteligencją.

Praca składa się z pięciu rozdziałów. Pierwszy zawiera opis architektury systemu. W drugim rozdziale zawarto dokumentację użytkownika systemu i szczegóły implementacji. Rozdział trzeci zawiera podsumowanie, rozdział czwarty podział prac a rozdział piąty spis płyty dołączonej do pracy.





# Rozdział 1

## Podstawowe pojęcia

### 1.1. Gra

Gra składa się z Reguł i Sędziego.

Reguły określają stan początkowy, dostępne graczom ruchy oraz warunki zwycięstwa, przegrannej, bądź remisu. Gra powinna być co najmniej dwuosobowa. W warunkach serwisu AI-Arena graczami będą najczęściej programy komputerowe, nazywane Botami bądź Programami Walczącymi.

Sędzia to program kontrolujący przebieg rozgrywki. Ma za zadanie:

- wyznaczyć stan początkowy każdej rozgrywki
- Odbierać komunikaty od programów grających, sygnalizujące ich zagrania
- Kontrolować poprawność zagrań graczy, oraz uaktualniać stan rozgrywki
- Informować Graczy o obecnym stanie rozgrywki
- Rozstrzygać czy gra się zakończyła i przydzielać punkty zwycięstwa graczom.

### 1.2. Program Walczący



## Rozdział 2

# Zastosowania w nauce i biznesie internetowym



## Rozdział 3

# Podobne platformy

Obecnie istnieją serwisy internetowe podobne do Ai-Arena. Oto kilka z nich:

### 3.1. Top Coder

Bardzo popularny serwis organizujący różnego rodzaju konkursy programistyczne. Jednym z nich są tzw Marathon Matche, podczas których uczestnicy wysyłają programy, które starają się najbardziej optymalnie rozwiązać dany problem, przy czym nie istnieje rozwiązanie całkowicie optymalne.

### 3.2. Ai Challenge

Serwis organizujący w sposób cykliczny zawody dla programów walczących. Najczęściej ok 2 konkursy rocznie. Każdy konkurs ma określony czas trwania i nie można w nim uczestniczyć po jego zakończeniu.

### 3.3. SPOJ

Serwis zawierający dużą bazę zadań algorytmicznych dostępnych do rozwiązywania użytkownikom. Zadania nie mają określonego terminu rozwiązywania. Istnieje ranking biorący pod uwagę liczbę wszystkich rozwiązanych przez użytkowników zadań.



## Rozdział 4

# Use cases

### Scenariusze użycia systemu przez użytkownika niezalogowanego (Gościa)

#### Dostęp do strony głównej

##### Założenia:

- Brak

##### Scenariusz postępowania:

1. Gość wpisuje w pasku adresu adres strony

##### Efekt:

1. W oknie przeglądarki wyświetla się strona główna

##### Scenariusze alternatywne:

- Brak

#### Rejestracja w systemie

##### Założenia:

1. Gość znajduje się na stronie głównej serwisu
2. Gość posiada aktywne konto mailowe

##### Scenariusz postępowania:

1. Gość klika na link ?zarejestruj? przekierowujący do strony z rejestracją
2. Gość wpisuje swoje dane do formularza. Dane zawierają m.in. pożądaną nazwę użytkownika, hasło, pole do weryfikacji hasła, adres mailowy.
3. Gość wchodzi na swoją skrzynkę mailową i otwiera list wysłany przez serwis
4. Gość klika w link aktywacyjny

##### Efekt:

1. System wysłał do użytkownika wiadomość na podany adres mailowy
2. System zakłada użytkownikowi nowe konto w serwisie
3. Po kliknięciu w link aktywacyjny system umożliwia zalogowanie na to konto

#### **Scenariusze alternatywne:**

1. Gość wybrał nazwę użytkownika, która jest już zajęta
  - a) System wyświetla komunikat, że żądana nazwa użytkownika jest niedostępna
2. Gość wpisał niejednakowe ciągi znaków w polu „hasło” i „potwierdzenie hasła”
  - a) System wyświetla komunikat o niezgodności danych
3. Gość wpisał niepoprawny adres mailowy
  - a) System próbuje wysłać maila na podany adres
  - b) W przypadku braku aktywacji konta w ciągu 24 godzin konto zostanie automatycznie skasowane
4. Gość nie kliknął w link aktywacyjny przysłany w mailu
  - a) Po upływie 24 godzin konto zostanie automatycznie skasowane

### **Logowanie w systemie - wariant 1: Gość posiada konto**

#### **Założenia:**

1. Gość znajduje się na stronie głównej serwisu
2. Gość posiada konto w serwisie

#### **Scenariusz postępowania:**

1. Gość klika w link ?zaloguj? przekierowujący na stronę logowania
2. Gość w formularzu wpisuje swoją nazwę użytkownika i hasło

#### **Efekt:**

1. System przekierowuje użytkownika na jego stronę startową

#### **Scenariusze alternatywne:**

1. Podana przez Gościa nazwa użytkownika jest nieprawidłowa
  - a) System wyświetla komunikat, że podana nazwa użytkownika lub hasło jest nieprawidłowe
2. Podane przez Gościa hasło jest nieprawidłowe
  - a) System wyświetla komunikat, że podana nazwa użytkownika lub hasło jest nieprawidłowe
3. Konto nie zostało aktywowane



- a) System wyświetla komunikat, że konto nie zostało jeszcze aktywowane. Użytkownik pozostaje niezalogowany

## **Logowanie w systemie - wariant 2: Gość nie posiada konta**

### **Założenia:**

1. Gość znajduje się na stronie głównej serwisu
2. Gość nie posiada konta w serwisie

### **Scenariusz postępowania:**

1. Gość klika w link ?zaloguj? przekierowujący na stronę logowania
2. Gość w formularzu wpisuje nazwę użytkownika i hasło

### **Efekt:**

1. System wyświetla komunikat, że podana nazwa użytkownika lub hasło są nieprawidłowe

### **Scenariusze alternatywne:**

- Brak



## Rozdział 5

# Metodyka Scrum



## Rozdział 6

# Architektura systemu

### 6.1. Nadzorca

Warstwa nadzorcy jest odpowiedzialna za uruchamianie rozgrywek pomiędzy wybranymi graczami, zbieranie informacji o ich wynikach i przekazywanie ich do warstwy schedulera. Nadzorca jest skryptem napisanym w pythonie, którego najważniejszą częścią jest metoda `play`. Metoda ta przyjmuje jako argumenty uruchamialne pliki sędziego i programów grających, oraz limity czasowy i pamięciowy dla każdego programu grającego. Następnie metoda `play` przeprowadza odpowiednią rozgrywkę, zwracając jako wynik ciąg liczb oznaczające przydzielone przez sędziego punkty za rozgrywkę, wykorzystane przez programy czasy i pamięć RAM.

### 6.2. Scheduler

Warstwa schedulera jest odpowiedzialna za ustalanie jakie rozgrywki w turnieju mają się odbyć, zlecanie poszczególnych rozgrywek nadzorcy i zapisywanie otrzymanych od nadzorcy wyników w bazie. Do realizacji warstwy nadzorcy używany jest program Gearmand<sup>0.26</sup>.

### 6.3. Baza danych

### 6.4. API użytkownika

Użytkownik komunikuje się z systemem poprzez interfejs webowy, zaimplementowany w Django. Interfejs pozwala na przesyłanie plików swoich programów grających...



## Rozdział 7

# Dokumentacja użytkowa i opis implementacji





## Rozdział 8

# Podsumowanie



## Rozdział 9

# Podział prac



## Rozdział 10

### Spis płyty



Dodatek A

Przykładowa gra





**Dodatek B**

**Przykładowe programy**



## **Dodatek C**

# **Przebieg przykładowego turnieju**



# Bibliografia