

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

**Jakub Tlałka**

Nr albumu: 292665

# **Implementacja systemu AI-Arena**

Praca licencjacka  
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem  
**dra. Janusza Jabłonowskiego**  
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. 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.

## **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>1. Wprowadzenie</b>	5
<b>2. Metodologia Pracy</b>	7
<b>3. Podstawowe pojęcia</b>	9
3.1. Gra	9
3.2. Program Walczący - Bot	9
3.3. Sędzia	9
3.4. Reguły	10
3.5. Mecz/Rozgrywka	10
3.6. Nadzorca	10
3.7. Gearman	10
3.8. Konkurs	10
<b>4. Zastosowania w nauce i biznesie</b>	11
<b>5. Podobne platformy</b>	13
5.1. Top Coder	13
5.2. Ai Challenge	13
5.3. SIO	13
5.4. SPOJ	13
5.5. MAIN	13
5.6. Tabela porównująca serwisy	14
<b>6. Architektura systemu</b>	15
6.1. Nadzorca	15
6.2. Scheduler	16
6.3. Serwis webowy	16
6.4. Baza danych	16
<b>7. Dokumentacja użytkowa i opis implementacji</b>	17
7.1. Scenariusze użycia systemu	17
7.1.1. Z udziałem użytkownika niezalogowanego (Gościa)	17
7.1.2. Z udziałem użytkownika zalogowanego	18
7.1.3. Z udziałem moderatora konkursu lub gry	22
<b>8. Podsumowanie</b>	25
<b>9. Podział prac</b>	27

<b>10.Spis płyty . . . . .</b>	<b>29</b>
<b>A. Przykładowa gra . . . . .</b>	<b>31</b>
<b>B. Przykładowe programy . . . . .</b>	<b>33</b>
<b>C. Przebieg przykładowego turnieju . . . . .</b>	<b>35</b>
<b>Bibliografia . . . . .</b>	<b>37</b>

# Rozdział 1

## Wprowadzenie

Serwis Ai-Arena to uniwersalna platforma do organizowania konkursów algorytmicznych. Główną domeną tych konkursów jest dziedzina sztucznej inteligencji, w szczególności inteligentnych programów walczących. Od parudziesięciu lat odbywają się zawody, mające na celu wyłonić programy najlepiej grające w popularne gry, takie jak szachy, warcaby itp. Serwis Ai-Arena pozwala organizować takie konkursy dla dowolnej gry wieloosobowej. Dodatkową zaletą jest istnienie rankingu stałego. W ten sposób można na bieżąco sprawdzać działanie własnych programów i mierzyć się z najlepszymi dotychczasowymi zawodnikami. Serwis może mieć zastosowania zarówno w nauce jak i w biznesie, ze względu na rosnące zapotrzebowanie na inteligentne programy rozwiązujące problemy w warunkach rywalizacji takich jak np giełda. Twórcy serwisu liczą na to, że wraz ze wzrostem jego popularności, Ai-Arena skupi społeczność ludzi zafascynowanych dziedziną sztucznej inteligencji i przyczyni się do rozwoju tej gałęzi nauki.





## Rozdział 2

# Metodologia Pracy

Podczas pracy nad systemem AI-Arena stosowaliśmy pewną modyfikację metodologii Scrum. Jest to metodologia typu lekkiego. Dzieli okres pracy na krótsze podokresy nazywane sprintami. Podczas każdego takiego sprintu, realizowany jest jakiś skończony kawałek funkcjonalności. Sprints nie powinny trwać dłużej niż dwa, trzy tygodnie. W naszym przypadku był to tydzień między kolejnymi spotkaniami na zajęciach z prowadzącym. Podczas takich zajęć, ustalany był zakres pracy przydzielony poszczególnym osobom na przyszły tydzień, oraz weryfikowana była praca wykonana w tygodniu poprzednim. Prowadzący zgłaszał uwagi do niektórych funkcjonalności. Dyskutowane były również dokumenty powstające przy okazji pracy nad projektem, przede wszystkim poniższa praca licencjacka. Zapis przebiegu zajęć, wraz z celami na przyszły tydzień, nazywany był minutkami i trzymany w Google Documents, tak by każdy z członków zespołu oraz prowadzący zajęć miał do niego dostęp.

Dodatkowo organizowane były spotkania samego zespołu. Odbywały się one również raz w tygodniu i miały na celu kontrolowanie przebiegu prac i zadbanie o to by zostały wykonane zaplanowane zadania. Spotkania odbywały się na wydziale, bądź jako konferencje video z wykorzystaniem Google Hangouts. Ich przebieg nadzorował Lider Projektu, który był odpowiedzialny również za realizowanie założeń przyjętej metodologii pracy.

Podział obowiązków między członkami zespołu był wynikiem porozumienia między członkami i był zatwierdzany przez lidera projektu, a następnie przedstawiany prowadzącemu zajęć. Do bieżącej kontroli stanu projektu użyty został system Redmine. Umożliwia on tworzenie zadań i przydzielanie ich członkom zespołu. Każde zadanie ma status, który zawiera się w zbiorze (otwarte, przydzielone, zrealizowane, zamknięte). Stan otwarty oznacza, że zadanie nie jest jeszcze przydzielone konkretnej osobie. Stan przydzielone oznacza, że zadanie jest przydzielone, a osoba odpowiedzialna za jego wykonanie jest w trakcie jego realizacji. Stan zrealizowane oznacza, że zadanie zostało wykonane przez osobę odpowiedzialną, ale nie zostało jeszcze zweryfikowane pod kątem poprawności. Stan zamknięte oznacza, że zadanie zostało zrealizowane i zweryfikowane.

Kod projektu, oraz dokumenty takie jak praca licencjacka, zostały umieszczone w repozytorium Gita. Git to system kontroli wersji dbający o archiwizację kodu oraz koordynację równoległych zmian wszystkich członków zespołu. Repozytorium jest publiczne i dostępne online w serwisie GitHub. Wszystkie zmiany dokonywane przez członków zespołu są archiwizowane, tak by można je było odtworzyć. Możliwe jest również jednoczesne rozwijanie projektu w kilku niezależnych kierunkach, a następnie połączenie efektów pracy.



## Rozdział 3

# Podstawowe pojęcia

### 3.1. Gra

Gra składa się z Reguł i Sędziego.

Reguły określają stan początkowy, dostępne graczom ruchy oraz warunki zwycięstwa, przegranej, bądź remisu.

Powinny ściśle określać Protokół Komunikacji między Programami Walczącymi, a Sędzią. 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.

### 3.2. Program Walczący - Bot

Jest to program napisany w jednym z obsługiwanych przez język serwisów. Musi być przypisany do konkretnej Gry dostępnej w serwisie. Uczestniczy w rozgrywkach (Meczach) z innymi botami przypisanymi do tej Gry. Ma za zadanie przetwarzać informacje o dotychczasowym przebiegu rozgrywki i produkować kolejne posunięcia, zgodne z regułami Gry.

### 3.3. Sędzia

Jest to program związany z daną Grą. Ma on za zadanie kontrolować przebieg Meczu, weryfikować zagrywki graczy, informować ich o aktualnym stanie, oraz ustalać wynik Meczu, zgodnie z zasadami związanymi z Grą, opisanymi w Regułach. Kod sędziego jest publiczny i dostępny wszystkim graczom.

### 3.4. Reguły

Dokument udostępniany publicznie, opisujący zasady gry, oraz protokół formatu komunikatów przesyłanych między Botami a Sędzią. Powinien być zgodny z implementacją programu Sędziego, oraz zapewniać możliwość przeprowadzenia dowolnej rozgrywki w obrębie zasad Gry.

### 3.5. Mecz/Rozgrywka

Jest rozgrywany między Botami, w obrębie konkursu lub poza nim. Jego przebieg jest kontrolowany przez Sędziego. Za przesyłanie komunikatów między Botami, a Sędzią, oraz odebranie od Sędziego rezultatu meczu odpowiedzialny jest Nadzorca.

### 3.6. Nadzorca

Program będący integralną częścią serwisu Ai-Arena. Jego zadaniem jest uruchamianie programów Botów i Sędziego podczas Meczu, a następnie transportowanie między nimi komunikatów tekstowych zgodnych z ustalonym protokołem. Nadzorca odbiera wynik Meczu od Sędziego i przekazuje go Gearmanowi, w celu zapisania rezultatu w bazie. Dodatkowo Nadzorca kontroluje czas i pamięć zużywaną przez Boty, tak by nie przekroczyły dozwolonych limitów. Informacje o ich zużyciu wraz z logiem rozgrywki generowanym przez sędziego, Nadzorca zwraca do Gearmana.

### 3.7. Gearman

Zewnętrzne narzędzie kolejkowania zadań i transportu danych. W Ai-Arena służy jako most między Nadzorcą, a Bazą danych. Kolejkuje Mecze do rozegrania, zleca ich rozegranie Nadzorcy, a następnie zapisuje do bazy przekazany przez Nadzorcę rezultat i dodatkowe informacje o przebiegu rozgrywki.

### 3.8. Konkurs

Konkurs może mieć określoną datę zakończenia, bądź być tzw konkursem stałym, w którym nie ma ostatecznego terminu wysyłania rozwiązań. Dla każdego konkursu tworzony jest ranking, bądź drabinka rozgrywek. Określają one kolejność botów, w szczególności zwycięzcę, bądź aktualnego lidera konkursu. W przypadku konkursu stałego istnieje ranking, który jest posortowany po sumie zdobytych przez bota punktów w meczach, przy czym każdy bot powinien mieć tę samą liczbę rozegranych meczy.

## Rozdział 4

# Zastosowania w nauce i biznesie

Sztuczna inteligencja jest jedną z szybciej rozwijających się obecnie dziedzin. Zastosowania algorytmów SI sięgają prawie wszystkich obszarów nie tylko internetu, ale i codziennego życia. Serwis AI-Arena pomaga rozwijać gałąź tej nauki związaną z rywalizacją.

Najprostsze przykłady rywalizacji to oczywiście wszelkiego rodzaju gry i sporty. Obecnie komputery są w stanie wygrywać z człowiekiem w większości gier takich jak szachy, warcaby itp. Coraz bardziej zaawansowani stają się przeciwnicy kierowani przez komputer w grach video. Również w sporcie zaczęto doceniać znaczenie metod naukowych do opracowywania optymalnych strategii. Prawdopodobnie kwestią czasu jest analizowanie gry wirtualnych zespołów kierowanych sztuczną inteligencją, a następnie wykorzystywanie obserwacji do poprawy gry prawdziwej drużyny.

Rywalizacja może być wykorzystana również jako metoda rozwiązywania problemów. Przykładem takiego podejścia są algorytmy genetyczne, w których najlepsze jednostki pozostają w obiegu, cały czas udoskonalając swoje podejście do rozwiązywania danego problemu.

Serwis AI-Arena ma duże zastosowanie w biznesie. Firmy nieustannie rywalizują między sobą w walce o klienta. Serwis umożliwia symulowanie takiej rywalizacji i dzięki temu odkrywanie skutecznych algorytmów sztucznej inteligencji, które będą podejmowały decyzje decydujące o sukcesie wykorzystującej je firmy.

Innym przykładem zastosowania AI-Arena są działania wojenne. Serwis może pomóc w szukaniu algorytmów, które będą potrafiły adaptować się do różnych warunków i w zależności od nich sugerować najlepsze strategie i taktyki w walce z przeciwnikiem.



## Rozdział 5

# Podobne platformy

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

### 5.1. 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.

### 5.2. 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.

### 5.3. SIO

Projekt od lat wykorzystywany do organizowania konkursów algorytmicznych, w szczególności polskiej Olimpiady Informatycznej, ale też Międzynarodowej Olimpiady Informatycznej. Jest to framework, który można wykorzystywać również w wersji lokalnej, nie tylko jako serwis internetowy.

### 5.4. 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.

### 5.5. MAIN

Posiada dużą bazę zadań z olimpiad informatycznych, ale także kursy umożliwiające pogłębienie wiedzy algorytmicznej.

## 5.6. Tabela porównująca serwisy

	tematyka	typ konkursów	baza zadań
Top Coder	algorytmy, problemy optymalizacyjne	okresowe	bardzo duża baza z przeszłych zawodów
Ai Challenge	programy walczące	okresowe	konkurs co pół roku
SPOJ	algorytmika	stałe/okresowe	duża baza zadań
Ai-Arena	programy walczące	stałe/okresowe	baza gier moderowana przez użytkowników
SIO	algorytmy	okresowe	większość zadań dostępna tylko podczas zawodów
MAIN	algorytmy	stałe	zadania z najważniejszych zawodów informatycznych



## Rozdział 6

# Architektura systemu

Na serwis AI-Arena składają się trzy warstwy: nadzorcy, schedulera i serwisu webowego.

### 6.1. Nadzorca

Nadzorca jest jądrem serwisu AI-Arena. 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 słownik zawierający:

- Ciąg liczb oznaczający przydzielone przez sędziego punkty za rozgrywkę
- Informacje na temat przebiegu rozgrywki
- Czas jaki zużyły programy walczące
- Pamięć RAM jaką wykorzystywały programy walczące

Rozgrywka zaczyna się przez uruchomienie programów walczących, oraz sędziego, kontrolującego przebieg rozgrywki. Komunikacja między sędzią a poszczególnymi programami odbywa się za pośrednictwem nadzorcy, który odczytuje komunikaty ze standardowego wyjścia programów i wypisuje informacje zwrotne na ich standardowe wejście. Format komunikatów od sędziego i botów powinien być wyspecyfikowany dla każdej gry, przy czym powinien on być zgodny z poniższym protokołem:

- Bot wysyła komunikaty TYLKO do sędziego (za pośrednictwem Nadzorcy)
- Każdy komunikat od Sędziego musi być potwierdzony komunikatem zwrotnym od Bota.
- Komunikaty wysyłane przez sędziego mają następujący format: zaczynają się od nawiasów kwadratowych, w których znajduje się lista graczy oddzielona przecinkami. Następnie znajduje się wiadomość, która zostanie przekazana odpowiednim botom. Komunikat musi kończyć się ciągiem `'«i' i znakiem nowej linii`, np: `'[1,2,3,4,5]INIT«i`
- Sędzia otrzymuje komunikaty zwrotne od botów w takiej kolejności w jakiej zostali wylistowani.
- Zakłada się że ciąg `'«i` kończy komunikat. Wysłanie komunikatu o nieprawidłowym formacie skutkuje zakończeniem rozgrywki.
- Komunikaty od nadzorcy do botów i do sędziego również kończą się ciągiem znaków `'«i`. Dzięki temu można przysyłać komunikaty wielowierszowe.
- Jeśli Sędzia chce wysłać komunikat do wszystkich może zacząć ALBO od wylistowania wszystkich graczy, ALBO użyć skrótu notacyjnego: `'[0]'` - to jeszcze nie działa, kwestia dopisania jednego `if'a`
- Ponadto, jeśli Sędzia stwierdził, że gra się zakończyła i chce poinformować wszystkich o tym, że nastąpił koniec gry wysyła do Nadzorcy komunikat o treści `"[0]END«i`. Następnie powinien wysłać komunikat zawierający punktację dla wszystkich graczy w postaci: `'[score1, score2, ... ]«i`
- Jeśli sędzia chce zakończyć działanie któregoś z graczy należy wysłać do niego wiadomość KILL (np. `"[4]KILL«i`). Można też w ten sposób zabić większą liczbę graczy lub nawet wszystkich (`"[0]KILL«i`)

- Jeśli sędzia wyśle komunikat do bota który odszedł w pokoju, zostanie komunikat zwrotny '\_DEAD\_'  
Boty powinny unikać wysyłania tego komunikatu, gdyż mogą zostać uznane za martwe.

## 6.2. Scheduler

Warstwa schedulera łączy część webową serwisu z nadzorcą. W ten sposób obie części funkcjonują niezależnie i mogą być używane jako osobne produkty. Scheduler jest kolejką zadań, w której trzymane są zlecenia rozegrania meczy. Zlecenia te są przekazywane nadzorcy, a następnie wyniki rozgrywek zwracane przez nadzorcę są zapisywane w bazie. Do realizacji warstwy nadzorcy używany jest program Gearmand0.26.

## 6.3. Serwis webowy

Użytkownik komunikuje się z systemem poprzez interfejs webowy, zaimplementowany w Django. Interfejs udostępni następujące akcje:

- Założenie konta
- Dodanie nowej gry do serwisu
- Wysłanie własnego programu walczącego w wybraną grę dostępną w serwisie
- Uruchomienie testowego meczu, między własnym programem, a innym wybranym programem, zgłoszonym do tej samej gry.
- Zgłoszenie programu walczącego do konkursu, gdzie będzie on rywalizował z pozostałymi zgłoszonymi programami.
- Przeglądanie zapisów rozgrywek.
- Przeglądanie aktualnego rankingu konkursu.
- Zapoznanie się z zasadami wybranego konkursu/gry
- Obejrzenie kodu źródłowego programu sędziego, dla danej gry.

## 6.4. Baza danych

Za obsługę bazy danych odpowiedzialny jest framework Django. Domyślną bazą, na której działa serwis jest PostgreSQL, ale nie jest to wymagane.

## Rozdział 7

# Dokumentacja użytkowa i opis implementacji

### 7.1. Scenariusze użycia systemu

#### 7.1.1. Z udziałem użytkownika niezalogowanego (Gościa)

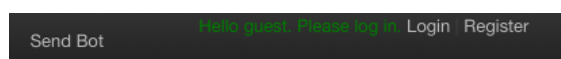
Użytkownik niezalogowany ma bardzo ograniczone prawa używania serwisu. Praktycznie jedyną dozwoloną aktywnością jest rejestracja w serwisie lub próba zalogowania się do utworzonego wcześniej konta.

##### Rejestracja w systemie

Aby zarejestrować się w systemie należy wykonać kilka kroków. Na początku należy kliknąć w napis "Register" zlokalizowany po prawej stronie na górze.

Następnie system przekieruje użytkownika do strony, gdzie będzie mógł wybrać swój login oraz hasło. Należy zwrócić uwagę, iż login musi być unikalny, może zawierać wyłącznie małe i duże litery, cyfry oraz znak '\_'. W przeciwnym wypadku pojawi się komunikat o błędzie.

Po prawidłowym wykonaniu tego kroku użytkownik zostanie przeniesiony na swoją stronę profilową. Może stąd zarządzać swoim kontem. Od tej chwili jest również zalogowany w serwisie, co znaczy, że może m. in. wysyłać swoje boty do gier lub wykonywać inne akcje dostępne wyłącznie dla zalogowanych użytkowników.



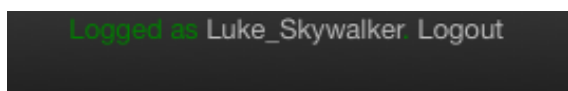
Rysunek 7.2: Pasek logowania i rejestracji do serwisu

## Logowanie

Logowanie jest standardową aktywnością, którą się spotyka na wszystkich forach i wielu serwisach. Przycisk logowania znajduje się tuż obok przycisku rejestracji. Aby się zalogować należy kliknąć w ten przycisk a następnie wypełnić pola formularza, który się pojawi na ekranie. Jeśli dane będą zgodne z tymi zapisanymi w bazie danych, system przekieruje użytkownika na jego stronę profilową.

### 7.1.2. Z udziałem użytkownika zalogowanego

#### Wylogowanie się z serwisu



Rysunek 7.3: Pasek wylogowania z serwisu

Wylogowanie jest kolejną standardową aktywnością udostępnianą przez wiele serwisów. Aby wylogować się z serwisu należy kliknąć przycisk "Logout", który znajduje się w prawym, górnym rogu strony. Zastępuje on przycisk "Login", który widoczny jest dla niezalogowanych użytkowników. Po kliknięciu tego przycisku użytkownik jest przekierowywany na stronę główną serwisu.

#### Otwarcie swojej strony profilowej

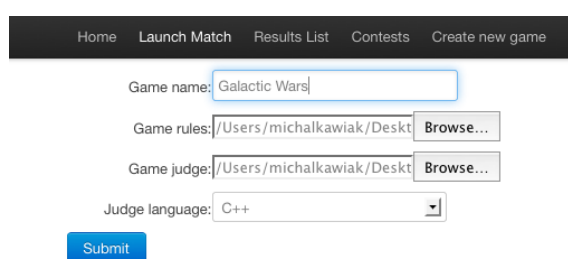
Aby otworzyć swoją stronę profilową użytkownik musi kliknąć w link znajdujący się pod jego loginem. Loginy użytkownika pojawiają się w różnych miejscach systemu i kliknięcie w dowolny z nich powoduje otwarcie strony profilowej danego użytkownika. W szczególności po kliknięciu na własny login pojawi się własna strona użytkownika. Istnieje również tzw. "stały link do profilu" znajdujący się na górze strony w prawym rogu. Po kliknięciu w znajdujący się tam login użytkownika (zaznaczonego innym kolorem niż reszta napisu) system wyświetla stronę profilową aktualnie zalogowanego użytkownika.

#### Edycja profilu użytkownika

Gdy użytkownik znajduje się na swojej stronie profilowej ma dostęp do kilku odnośników. Jednym z nich jest odnośnik do strony konfiguracji-

nej profilu. Aby ją otworzyć należy kliknąć w link "Edit profile" znajdujący się pod zdjęciem profilowym. W oknie edycji profilu pojawia się formularz zawierający kilka pól, m.in. zdjęcie, kraj, szkoła/universytet czy zainteresowania. Żadne z nich nie jest obowiązkowe. Użytkownik zatwierdza zmiany klikając na przycisk "Update" znajdujący się na samym dole formularza.

### Utworzenie nowej gry lub konkursu



Rysunek 7.4: Formularz do tworzenia nowej gry

Tworzenie gry (konkursu) jest jedną z centralnych aktywności w serwisie. Aby utworzyć grę, użytkownik musi mieć wcześniej przygotowane następujące pliki:

- Plik źródłowy sędziego.
- Plik z zasadami rozgrywki

Sędzia jest programem, który nadzoruje przebieg gry. Powinien być napisany w jednym z języków: C, C++, Java, Python. Otrzymuje on komunikaty od Nadzorcy, który

z kolei czytuje je ze standardowego wejścia Botów. Sędzia musi dokonać walidacji komunikatu, tj. upewnić się, iż format komunikatu jest zgodny z protokołem opisanym w zasadach gry oraz, że ruch zawarty w komunikacie nie narusza reguł gry (przykładowo, że gracz grający pionami czarnymi nie ruszył się białym pionem). Gdy upewni się, że otrzymany komunikat jest poprawny wysyła na swoje standardowe wyjście komunikat zgodnie z formatem opisanym w regułach gry.

Plik źródłowy sędziego jest dostępny dla wszystkich zalogowanych użytkowników serwisu. Ma to zapobiec wysyłaniu "stronniczych" sędziów oraz łatwieszemu ujawnieniu pewnych "ukrytych" wad protokołu. Administratorzy zastrzegają sobie prawo do usunięcia lub poprawienia wadliwych sędziów.

Plik z zasadami rozgrywki musi zawierać ogólny opis gry (np. fabułę wprowadzającą) oraz szczegółowy opis i znaczenie komunikatów. W szczególności muszą być zawarte informacje takie jak:

- dokładny format komunikatu

- dozwolone odpowiedzi na każdy rodzaj komunikatu
- komunikaty informujące o błędzie

Zostało przyjęte założenie, że Bot, który wysła komunikat o innej, niż wyspecyfikowano formie automatycznie przegrywa daną grę i jest zabijany przez Nadzorcę.

Aby utworzyć konkurs nie jest potrzebne wcześniejsze przygotowywanie plików, jednak musi istnieć gra, na bazie której chcemy utworzyć konkurs.

Gdy użytkownik przy-

gotował już powyż-

sze pliki powinien,

w celu utworzenia

nowej gry, kliknąć

w link "Create New

Game" znajdujący

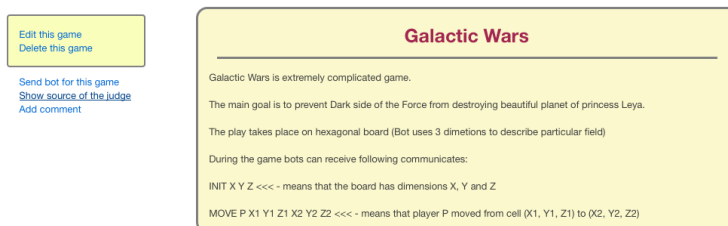
się na górnym pa-

sku strony. Otworzy

się formularz zawie-

rający pola, do któ-

rych należy wprowadzić informacje nt. tworzonej gry (między innymi 2 wymienione wcześniej pliki czy nazwa gry). Warto zwrócić uwagę, iż nazwa gry musi być unikalna, gdyż definiuje ona ścieżkę do poszczególnych plików na dysku. Gdy użytkownik wypełni formularz i zatwierdzi przyciskiem "Submit" na dole formularza zostanie przeniesiony na stronę zawierającą detale właśnie utworzonej gry. Autor staje się automatycznie moderatorem gry i ma prawo m. in. do edycji czy usunięcia gry lub edycji i usuwania komentarzy użytkowników.



Rysunek 7.5: Szczegóły gry

## Wysłanie bota do gry lub konkursu

Aby wysłać bota do gry (konkursu) należy w pierwszej kolejności przejść na stronę zawierającą szczegółowy opis gry. Można to zrobić klikając na przycisk "Games list" ("Conte-

Rysunek 7.6: Formularz do wysyłania bota

sts”) znajdujący się na górze strony, a następnie wy-

brać grę (konkurs) z wyświetlonej listy.

Gdy znajdujemy się już na stronie z detalami należy, w celu zgłoszenia bota, kliknąć na link "Send bot for this game" (lub "Add contestant" w przypadku konkursu) i wypełnić formularz wyświetlony przez system. Podobnie, jak w przypadku tworzenia gry należy mieć uprzednio przygotowany plik zawierający kod źródłowy bota. Bot komunikuje się z sędzią za pomocą komunikatów opisanych w zasadach gry wypisywanych i czytanych ze standardowego wejścia i wyjścia. W przypadku, gdy bot wyśle źle sformatowany lub błędny komunikat - automatycznie przegrywa rozgrywkę dostając za nią możliwe minimum punktów.

### Dodanie komentarza pod grą lub konkursem

Aby dodać komentarz do gry (konkursu) należy znaleźć się na stronie ze szczegółowym opisem danej gry (konkursu). Następnie kliknąć w link "Add comment" i wypełnić formularz wyświetlony przez system. Komunikat ten pojawi się w kolejności odwrotnie chronologicznej pod opisem gry.



Rysunek 7.7: Komentarz umieszczony pod grą

prywatność itp.) moderator konkursu ma prawo do edycji lub usunięcia komentarza. Takie samo prawa ma również autor komentarza.

### Wyświetlenie kodu źródłowego sędziego

```
1 /* quick and dirty hack to grab all credentials in the cred hash table
2  * from kernel via sysctl.
3  * sysctl is only defined if xnu is built with DEBUG_CRED defined.
4  */
5
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <fcntl.h>
9 #include <limits.h>
10 #include <string.h>
11 #include <errno.h>
12 #include <unistd.h>
13 #include <sys/stat.h>
14 #include <sys/types.h>
15 #include <sys/sysctl.h>
16 #include <bsm/audit.h>
17
18 /* bad! this is replicated in kern_credential.c. make sure they stay in sync!
19  * Or better yet have common header file?
20  */
21 struct debug_ucred {
22     uint32_t credp;
23     uint32_t cr_ref; /* reference count */
24     uid_t cr_uid; /* effective user id */
25 }
```

W celu wyświetlenia kodu źródłowego sędziego należy znajdować się na stronie gry za-

wiejszącej szczegółowy jej opis. Potem, po kliknięciu w link "Show source of the judge" system przekieruje nas na stronę, gdzie wy-

świetlony zostanie kod źródłowy sędziego.

### **7.1.3. Z udziałem moderatora konkursu lub gry**

#### **Usunięcie komentarza do gry lub konkursu**

Żeby móc usunąć komentarz użytkownik musi spełniać (minimum) jeden z poniższych warunków:

1. Być autorem komentarza
2. Być moderatorem gry (konkursu), pod którym został ten komentarz dodany
3. Być administratorem strony

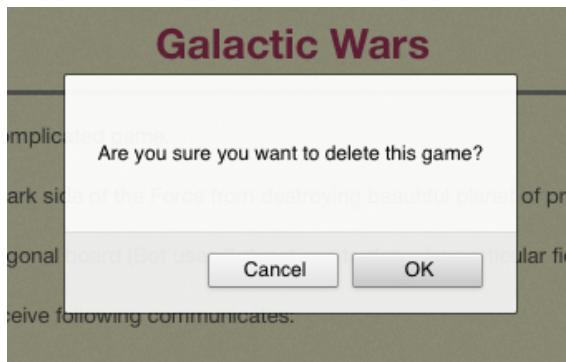
Jeśli wymaganie wstępne jest spełnione, wystarczy kliknąć w przycisk "Delete" znajdujący się w prawym górnym rogu komentarza. Pojawi się wtedy okienko z prośbą o potwierdzenie swojej decyzji. Należy pamiętać, że usuniętego komentarza nie można przywrócić! Wciśnięcie przycisku "OK" na pojawiającym się okienku spowoduje usunięcie komentarza i powrót do strony z detalami gry (konkursu)

#### **Edycja komentarza do gry lub konkursu**

Edycja komentarza jest podobna do dodawania nowego komentarza. Aby edytować istniejący komentarz należy kliknąć w link "Edit" znajdujący się w górnej części komentarza. Pojawi się wtedy formularz analogiczny do tego, który się pojawia przy dodawaniu komentarza. Po zatwierdzeniu dokonywanych zmian poprzez naciśnięcie przycisku "Save" zostaniemy przekierowani do strony ze szczegółowymi informacjami o grze. Zauważmy, że edytowany komentarz jest już wyświetlany w zmienionej formie.

#### **Usunięcie gry lub konkursu**





Rysunek 7.9: Monit o potwierdzenie usunięcia gry

to decyzja nieodwracalna! Potwierdzenie wykonania operacji spowoduje usunięcie gry z listy dostępnych i przeniesienie na stronę główną serwisu.

#### Edycja gry lub konkursu



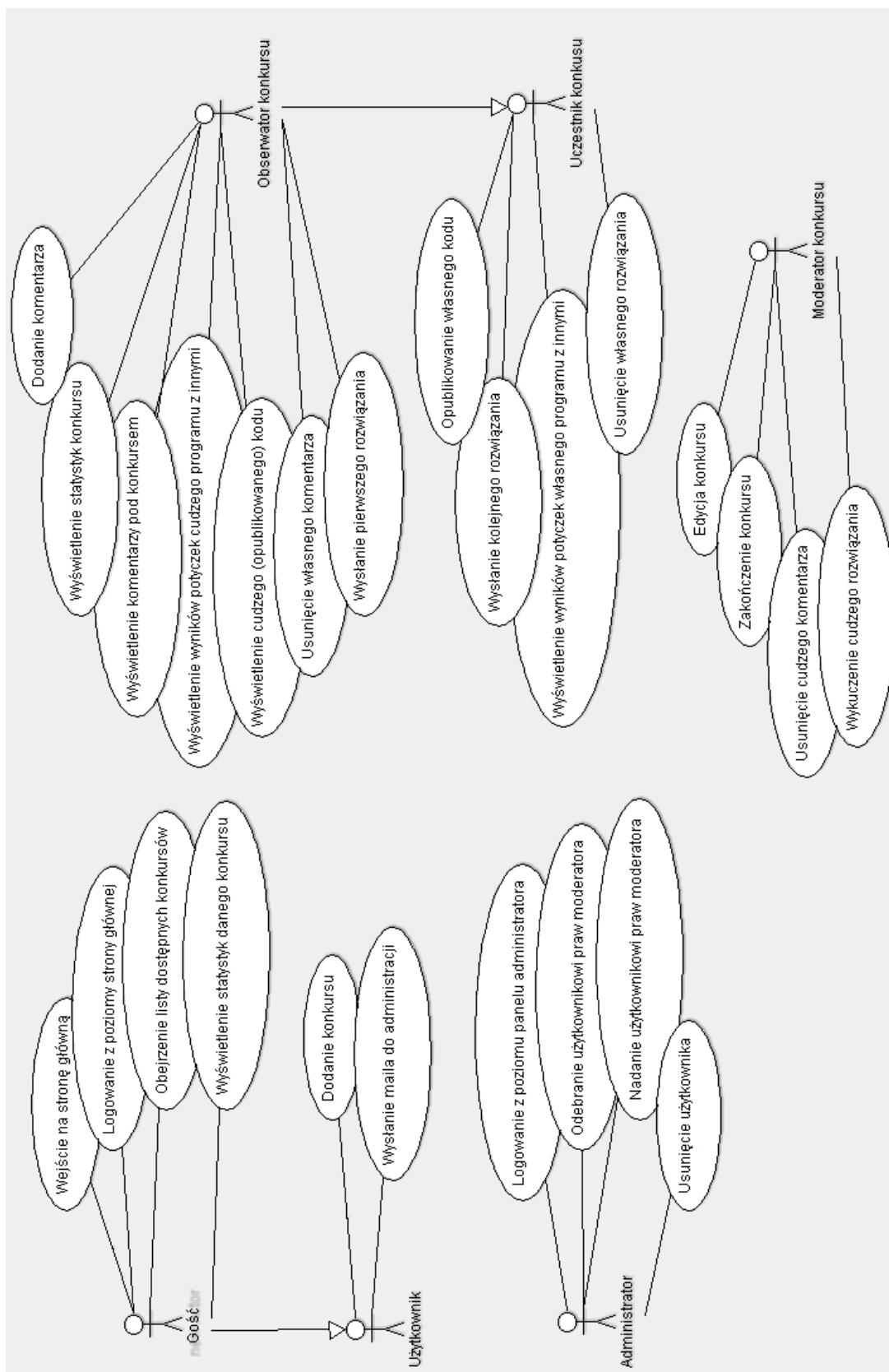
Rysunek 7.10: Sposób wyświetlania kodu źródłowego

Aby móc usunąć grę (konkurs) trzeba spełniać jeden z warunków:

1. Być moderatorem gry
2. Być administratorem strony

W celu usunięcia gry należy znajdować się na stronie zawierającej szczegółowy opis gry. Następnie kliknąć w znajdujący się po lewej stronie link "Delete this game". Spowoduje to wyskoczenie okienka z prośbą o potwierdzenie swojej decyzji. Pamiętajmy, iż jest

Możliwa jest edycja konkursu. Aby tego dokonać trzeba być administratorem lub moderatorem gry. Należy kliknąć w link "Edit game". System przeniesie nas do okna, w którym wyświetli się formularz do edycji gry. Po wprowadzeniu zmian należy zatwierdzić przyciskiem "Save".



Rysunek 7.1: Diagram przypadków użycia

## 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