

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

1. Wprowadzenie	5
2. Metodologia Pracy	7
3. Podstawowe pojęcia	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
4. Zastosowania w nauce i biznesie	11
5. Podobne platformy	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
6. Architektura systemu	15
6.1. Nadzorca	15
6.2. Scheduler	16
6.3. Serwis webowy	16
6.4. Baza danych	16
7. Dokumentacja użytkowa i opis implementacji	19
7.1. Scenariusze użycia systemu	19
7.1.1. Z udziałem użytkownika niezalogowanego (Gościa)	19
7.1.2. Z udziałem użytkownika zalogowanego	20
7.1.3. Z udziałem moderatora konkursu lub gry	24
8. Podsumowanie	27
9. Podział prac	29

10.Spis płyty	31
A. Przykładowe gry	33
A.1. Szachy	33
A.2. Poker	33
A.3. Gra giełdowa	33
B. Przykładowe programy	35
C. Przebieg przykładowego turnieju	37
Bibliografia	39

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, ale pozwala również na organizację konkursów algorytmicznych, włącznie z zadaniami interaktywnymi, co sprawia, że istnieje pełna dowolność wyboru rodzaju konkursu. 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. Projekt jest udostępniony w serwisie Github. Każda osoba chętna do pomocy przy jego rozwoju może uzyskać prawa zapisu, wystarczy skontaktować się z twórcami i poprosić o dostęp.

Rozdział 2

Metodologia Pracy

Aby usprawnić proces tworzenia systemu AI-Arena, postanowiliśmy na początku prac nad systemem wybrać odpowiednią do naszych potrzeb metodologię pracy. Początkowo rozważaliśmy następujące metodologie: Agile Uniformed Process, Extreme Programming, Scrum. Przy dokonywaniu wyboru kierowaliśmy się następującymi kryteriami:

- Ponieważ pracowaliśmy w zespole czteroosobowym, wybrana metodyka powinna być metodyką lekką.
- Ponieważ spotykaliśmy się co tydzień na proseminarium, chcieliśmy aby podstawową jednostką pracy były tygodniowe iteracje.
- Ponieważ w trakcie tworzenia projektu wszyscy członkowie zespołu kontynuowali studia i musieli wypełniać związane z tym obowiązki, chcieliśmy aby każdy z członków mógł samodzielnie podjąć decyzję o ilości pracy jaką wykona w najbliższym tygodniu.
- Praca nad projektem miała przebiegać w sposób rozproszony. Chcieliśmy mieć możliwość pracy w dogodnych dla siebie godzinach, niezależnie od pozostałych członków zespołu.
- Wybrana przez nas metodyka powinna również pozwalać na łatwe śledzenie zmian w projekcie, oraz ułatwiać zapewnianie kontroli jakości.

Żadna z początkowo rozważanych metodologii nie spełniała wszystkich naszych wymagań. Zdecydowaliśmy się więc na modyfikację metodologii Scrum. Jest to metodologia typu lekkiego. Dzieli okres pracy na krótsze podokresy nazywane sprintami. Podczas każdego takiego sprintu, realizowana jest skończona część 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 zajęcia 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 w budynku Wydziału Matematyki Informatyki

i Mechaniki UW, 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 członków 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 serwis języków. 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. Warto wspomnieć mecz szachowy mistrza świata Garriego Kasparowa z komputerem Deep Blue czy teleturniej Va Banque z udziałem komputera Watson - w obu przypadkach zwyciężyła sztuczna inteligencja (http://www.benchmark.pl/aktualnosci/Superkomputer_IBM_wystapil_w_teleturnieju-32781.html). Coraz bardziej zaawansowani stają się przeciwnicy kierowani przez komputer w grach video. Również w sporcie zaczęto doceniać znaczenie metod naukowych do opracowywania optymalnych strategii. Z powodzeniem wykorzystuje się komputer do badania statystyk w amerykańskiej lidze baseballowej (<http://research.sabr.org/journals/online/38-brj-1976/165-computers-in-baseball-analysis>). 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. Można także symulować inwestowanie funduszy np. na giełdzie. Banki wykorzystują sztuczną inteligencję do organizacji operacji finansowych i zarządzania środkami. Sztuczna inteligencja odnosi już sukcesy w rywalizacji z ludźmi w symulacji obrotów finansowych (<http://news.bbc.co.uk/2/hi/business/1481339.stm>).

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:

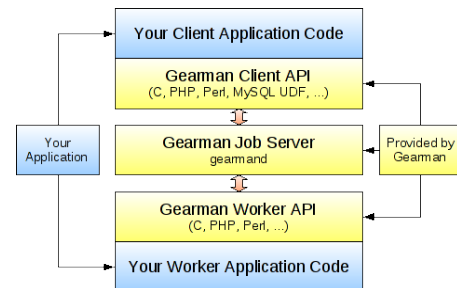
- Bot wysyła komunikaty TYLKO do sędziego (za pośrednictwem Nadzorcy)
- Każdy komunikat od Sędziego musi być potwierdzony komunikatem zwrotnym od Bota.
- Komunikaty wysyłane przez sędziego mają następujący format: zaczynają się od nawiasów kwadratowych, w których znajduje się lista graczy oddzielona przecinkami. Następnie znajduje się wiadomość, która zostanie przekazana odpowiednim botom. Komunikat musi kończyć się ciągiem `'«|'` i znakiem nowej linii, np: `'[1,2,3,4,5]INIT«|newline'`.
- Sędzia otrzymuje komunikaty zwrotne od botów w takiej kolejności w jakiej zostali wylistowani.
- Zakłada się że ciąg `'«|newline'` kończy komunikat. Wysłanie komunikatu o nieprawidłowym formacie skutkuje zakończeniem rozgrywki.
- Komunikaty od nadzorcy do botów i do sędziego również kończą się ciągiem znaków `'«|newline'`. 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«|newline"`. Następnie powinien wysłać komunikat zawierający punktację dla wszystkich graczy w postaci: `'[score1, score2, ...]«|newline'`.
- Jeśli sędzia chce zakończyć działanie któregoś z graczy należy wysłać do niego wiadomość KILL (np. `"[4]KILL«|newline"`). Można też w ten sposób zabić większą liczbę graczy lub nawet wszystkich (`"[0]KILL«|newline"`)

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

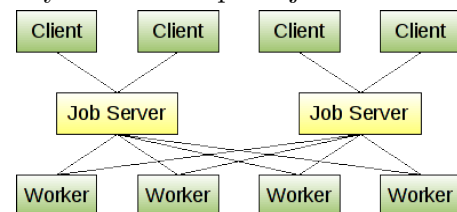
6.2. Scheduler

Do realizacji warstwy Schedulera używany jest program Gearmand0.26. Gearman jest lekkim frameworkiem służącym do przydzielania zadań do wykonania określonym maszynom lub procesom. Posiada prosty interfejs i jest łatwy w użyciu. Każda aplikacja korzystająca z tego frameworku składa się z trzech części: klienta, workera oraz serwera. Rolę klienta pełni część webowa serwisu, która zleca rozegranie odpowiedniej Rozgrywki, podając jako argumenty grę, bota i sędziego Rozgrywki. To zadanie jest zgłaszane do serwera, który je kolejkuje oraz przydziela workerowi. Zadanie workera ogranicza się do uruchomienia nadzorca z odpowiednimi parametrami, a po zakończeniu Rozgrywki zapisaniu wyników do bazy danych.

Warstwa Schedulera umożliwia zakolejkowanie kilku Meczów do rozegrania np. w razie dużego ruchu w serwisie lub gdy wymaga tego organizacja Turnieju, oraz wykonanie tych zadań w sposób asynchroniczny. Gearman łączy część webową serwisu z nadzorcą, przez co obie części funkcjonują i mogą być rozwijane niezależnie. Umożliwia także użycie wielu klientów, serwerów i workerów tworzących klastery. Użycie zapasowego serwera pomogłoby ustabilizować pracę serwisu, a uruchomienie kilku workerów na wielordzeniowej maszynie znacznie ją przyspieszyć.



Rysunek 6.1: Aplikacja Gearmana



Rysunek 6.2: Klastery Gearmana

6.3. Serwis webowy

Użytkownik komunikuje się z systemem poprzez interfejs webowy, zaimplementowany w Django. Interfejs udostępnia 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/grzy
- Obejrzenie kodu źródłowego programu sędziego, dla danej gry.

6.4. Baza danych

Za obsługę bazy danych odpowiedzialny jest framework Django. Współpracuje z bazami: PostgreSQL, MySQL, SQLite, Microsoft SQL Server oraz Oracle. Domyślną bazą, na której działa serwis jest PostgreSQL, ale nie jest to wymagane. Tabele w bazie danych są tworzone przez Django w sposób automatyczny na podstawie modeli w pliku models.py, każdemu modelowi odpowiada jedna tabela w bazie. Oprócz tego istnieje plik fixture, do którego można zapisać lub wczytać z niego dane znajdujące się w bazie. Służy on do tworzenia testowej bazy danych.

W serwisie są użyte następujące modele(wraz z najważniejszymi polami):

Game

- nazwa
- ścieżka do programu sędziego
- plik z zasadami

Bot

- nazwa
- właściciel
- plik źródłowy

Ranking

- typ ranking - od niego zależy w jaki sposób będzie rozgrywany konkurs

BotRanking - przechowuje opis pozycji Bota w danym Ranking Contest

- nazwa
- Gra
- Ranking
- zawodnicy - Boty, które biorą udział w konkursie

Match

- Bot
- wynik
- wykorzystany czas
- zużyta pamięć

MatchBotResult - zawiera rezultat Bota w pojedynczym Meczu

- Game
- Contest
- wyniki uczestniczących Botów

UserProfile

- User
- podstawowe dane: zdjęcie, państwo itp.

UserNews

GameComment - komentarz do Gry

ContestComment - komentarz do Konkursu

Rozdział 7

Dokumentacja użytkowa i opis implementacji

7.1. Scenariusze użycia systemu

7.1.1. Z udziałem użytkownika niezalogowanego (Gościa)

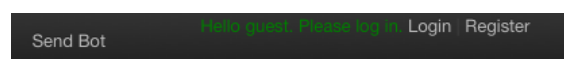
Użytkownik niezalogowany ma bardzo ograniczone prawa używania serwisu. Praktycznie jedyną dozwoloną aktywnością jest rejestracja w serwisie lub próba zalogowania się do utworzonego wcześniej konta.

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.



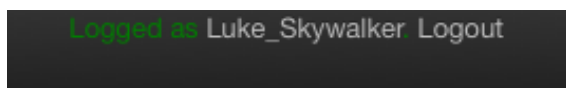
Rysunek 7.2: Pasek logowania i rejestracji do serwisu

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

- 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

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

Rysunek 7.4: Formularz do tworzenia nowej gry

- 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 nowy konkurs.

Gdy

użytkownik przygotował już powyższe pliki powinien, w celu utworzenia nowej gry, kliknąć w link "Create New Game" znajdujący się na górnym pasku strony. Otworzy się formularz zawie-

Edit this game
Delete this game

[Send bot for this game](#)
[Show source of the judge](#)
[Add comment](#)

Galactic Wars

Galactic Wars is extremely complicated game.

The main goal is to prevent Dark side of the Force from destroying beautiful planet of princess Leya.

The play takes place on hexagonal board (Bot uses 3 dimetions to describe particular field)

During the game bots can receive following communicates:

INIT X Y Z <<< - means that the board has dimensions X, Y and Z

MOVE P X1 Y1 Z1 X2 Y2 Z2 <<< - means that player P moved from cell (X1, Y1, Z1) to (X2, Y2, Z2)

Rysunek 7.5: Szczegóły gry

rający pola, do których 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.

Wysłanie bota do gry lub konkursu

Bot name:

Bot source:

Bot language:

Rysunek 7.6: Formularz do wysłania bota

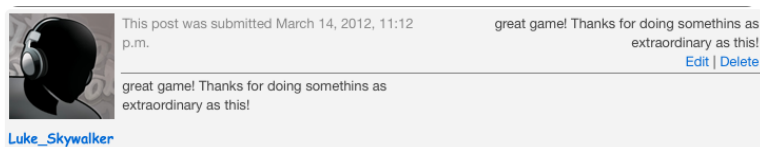
Aby wysłać bota do gry (konkursu) należy w pierwszej kolejności przejść na stronę zawierającą szczegółowy opis gry. Można to zrobić klikając na przycisk "Games list" ("Contests") znajdujący się na gó-

rze strony, a następnie wybrać 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 czytywanych 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

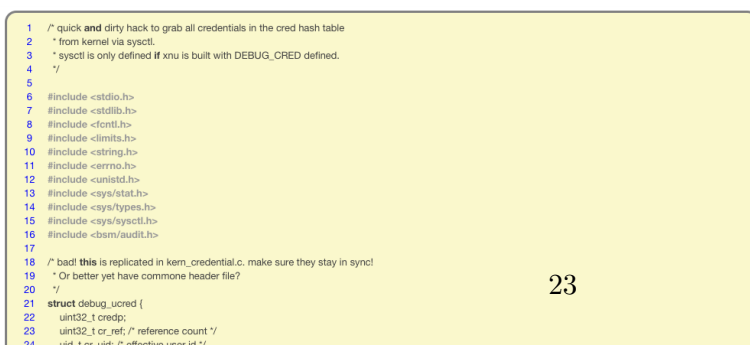
Aby dodać komentarz do gry (konkursu) należy znaleźć się na stronie ze szczegółowym opisem danej gry (konkursu). Następnie kliknąć w link "Add comment" i wypełnić formularz wyświetlony przez system. Komunikat ten pojawi się w kolejności odwrotnie chronologicznej pod opisem zasad gry.



Rysunek 7.7: Komentarz umieszczony pod grą

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

Wyświetlenie kodu źródłowego sędziego



Rysunek 7.8: Sposób wyświetlania kodu źródłowego

Należy zwrócić uwagę, iż w przypadku naruszania zasad publicznego wypowiedzania się (treści wulgarne, obraźliwe, naruszające prywatność itp.) moderator konkursu ma prawo do edycji lub usunięcia komentarza. Takie samo prawa ma również autor komentarza.

W celu wyświetlenia kodu źródłowego sędziego należy znaleźć się na stronie gry zawierającej szczegó-

łowy jej opis. Po-
tem, po kliknięciu
w link "Show sour-
ce of the judge" sys-
tem 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

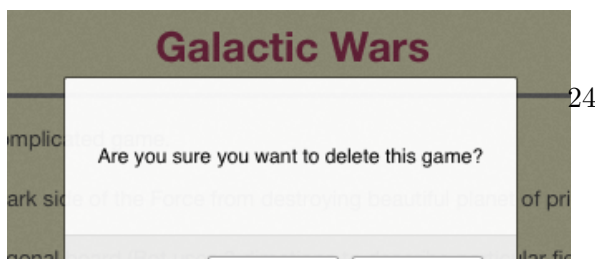
Jeśli wymaganie wstępne jest spełnione, wystarczy kliknąć w przycisk "Delete" znajdujący się w prawym górnym rogu komentarza. Pojawi się wtedy okienko z prośbą o potwierdzenie swojej decyzji. Należy pamiętać, że usuniętego komentarza nie można przywrócić! Wciśnięcie przycisku "OK" na pojawiającym się 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

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



1. Być moderatorem gry
2. Być administratorem strony

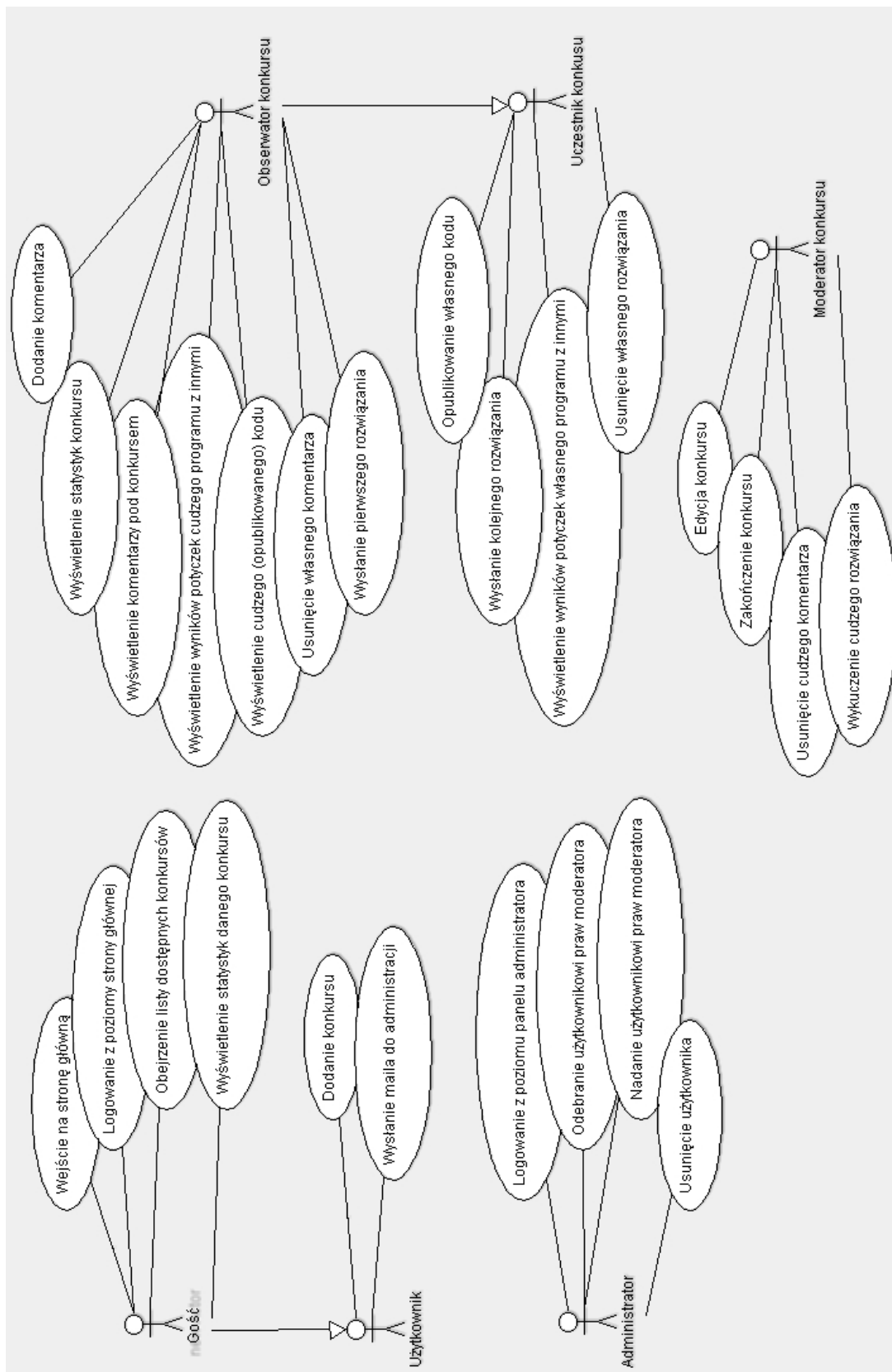
W celu usunięcia gry należy znajdować się na stronie zawierającej szczegółowy opis gry. Następnie kliknąć w znajdujący się po lewej stronie link "Delete this game". Spowoduje to wyskoczenie okienka z prośbą o potwierdzenie swojej decyzji. Pamiętajmy, iż jest to decyzja nieodwracalna! Potwierdzenie wykonania operacji spowoduje usunięcie gry z listy dostępnych i przeniesienie na stronę główną serwisu.

Edycja gry lub konkursu



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 7.10: Sposób wyświetlania kodu źródłowego



Rysunek 7.1: Diagram przypadków użycia

Rozdział 8

Podsumowanie

Rozdział 9

Podział prac

Rozdział 10

Spis płyty

Dodatek A

Przykładowe gry

A.1. Szachy

Znana na całym świecie gra dwuosobowa, rozgrywana na kwadratowej planszy o 64 polach. Gracze wykonują naprzemienne posunięcia figurami, a gra kończy się remisem, bądź zwycięstwem któregoś z graczy. Gra jest całkowicie deterministyczna, ale jej złożoność nie pozwoliła dotychczas na znalezienie uniwersalnej strategii wygrywającej, bądź nieprzegrywającej. Jednak obecne programy są już od wielu lat w stanie wygrać z najlepszymi graczami ludzkimi.

A.2. Poker

Gra wieloosobowa, zawierająca elementy hazardu. Rozgrywka zaczyna się od losowego przydziału kart graczom, dlatego nie istnieje jedyna optymalna strategia grania. Celem gry jest zdobycie wszystkich punktów posiadanych przez innych graczy, które są potrzebne, by uczestniczyć w kolejnych rozdaniach. Istnieje wiele wariantów Pokera, najbardziej znanym jest tzw Texas Hold'em.

A.3. Gra giełdowa

Każdy gracz ma początkowo określone fundusze. Następnie inwestuje on w akcje dostępne na giełdzie. W zależności od inwestycji innych graczy, zmienia się wartość akcji, a co za tym idzie, dorobek każdego z graczy. Celem jest oczywiście powiększenie swojego kapitału o jak największą sumę, w kolejnych rundach.

Dodatek B

Przykładowe programy

Dodatek C

Przebieg przykładowego turnieju

Bibliografia