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 implementacje 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 walczace, 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

Rozdział 1

Wprowadzenie

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. Sprinty 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 przydzielona, 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, sygnalizujace 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 pomost 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 udoskanalając 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 pomóć 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

	Top Coder	Ai Challenge	SPOJ	AI-Arena
Tematyka	algorytmy problemy optymalizacyjne	programy walczące	algorytmika	programy walczące
Typ konkursów	okresowe	okresowe	stałe/okresowe	stałe/okresowe
Różnorodność	regularne i częste konkursy w różnych kategoriach	konkurs co pół roku	duża baza zadań	baza gier modyfikowana przez użytkowników

Rozdział 6

Architektura systemu

Na serwis AI-Arena składają się trzy wartwy: 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 przekazywnie 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 sedziego 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 ktorych znajduje się lista garczy oddzielona przecinkami. Następnie znajduje się wiadomość, która zostanie przekazana odpowiednim botom. Komunikat musi kończy się ciągiem '«¡' i znakiem nowej linii, n.p: '[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 przesyłać 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, kewstia 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, dostanie komunikat zwrotny '_DEAD_«¡newline'. 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 udostępnie 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

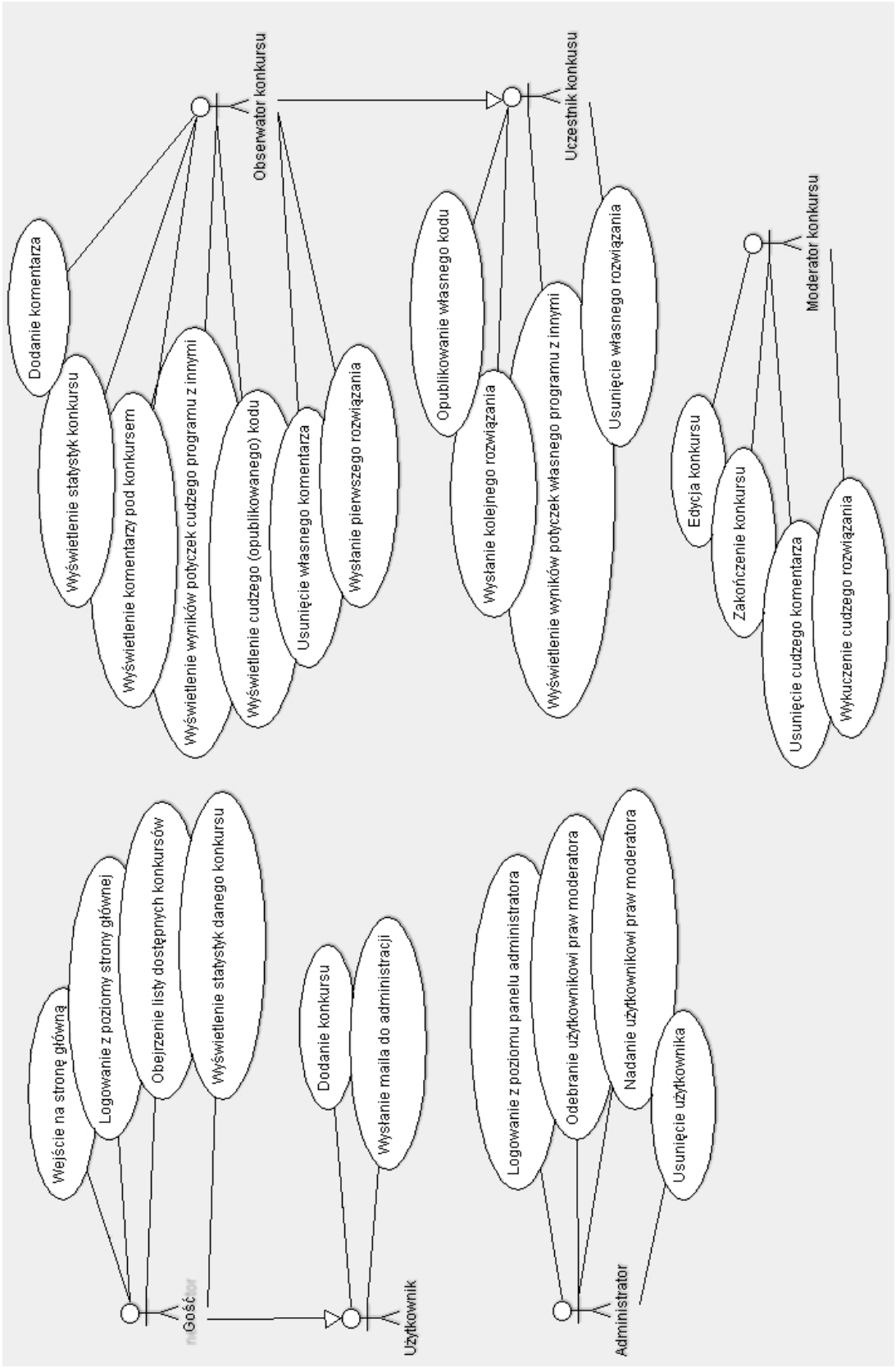
Dokumentacja uzytkowa i opis implementacji

7.1. Scenariusze użycia systemu

7.1.1. Z udziałem użytkownika niezalogowanego (Gościa)

7.1.2. Z udziałem użytkownika zalogowanego

7.1.3. Z udziałem moderatora konkursu lub gry



Rysunek 7.1: Diagram przypadków użycia

Rozdział 8

Podsumowanie

Rozdział 9

Podział prac

Rozdział 10

Spis płyty

Dodatek A

Przykladowa gra

Dodatek B

Przykladowe programy

Dodatek C

Przebieg przykladowego turnieju

Bibliografia