

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

Wprowadzenie	5
1. Podstawowe pojęcia	7
1.1. Gra	7
1.2. Konkurs	7
1.3. Program Walczący - Bot	7
2. Zastosowania w nauce i biznesie	9
3. Podobne platformy	11
3.1. Top Coder	11
3.2. Ai Challenge	11
3.3. SPOJ	11
3.4. Tabela porównująca serwisy	11
4. Use cases	13
5. Metodyka Scrum	17
6. Architektura systemu	19
6.1. Nadzorca	19
6.2. Scheduler	20
6.3. Serwis webowy	20
6.4. Baza danych	21
7. Dokumentacja użytkowa i opis implementacji	23
8. Podsumowanie	25
9. Podział prac	27
10. Spis płyty	29
A. Przykładowa gra	31
B. Przykładowe programy	33
C. Przebieg przykładowego turnieju	35
Bibliografia	37

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, 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.

1.2. 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.

1.3. 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.

Rozdział 2

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 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.

3.4. 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 przez użytkowników	baza gier modyfikowana

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

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 garczy oddzielona przecinkami. Następnie znajduje się wiadomość, która zostanie przekazana odpowiednim botom. Komunikat musi kończyć 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 '«\n' kończy komunikat. Wysłanie komunikatu o nieprawidłowym formacie skutkuje zakończeniem rozgrywki.
- Komunikaty od nadzorcy do botów i do sędziego również kończą się ciągiem znaków '«\n'. 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«\n". Następnie powinien wysłać komunikat zawierający punktację dla wszystkich graczy w postaci: '[score1, score2, ...]«\n'.
- Jeśli sędzia chce zakończyć działanie któregoś z graczy należy wysłać do niego wiadomość KILL (np. "[4]KILL«\n"). Można też w ten sposób zabić większą liczbę graczy lub nawet wszystkich ("[0]KILL«\n")
- 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

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

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