



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Sztuczna inteligencja w grach [S2SI1E>SIG]

Przedmiot

Kierunek studiów

Sztuczna inteligencja/Artificial Intelligence

Rok/Semestr

2/3

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

angielski

Forma studiów

stacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

20

Laboratorium

20

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

2,00

Koordynatorzy

dr inż. Paweł Wojciechowski

pawel.wojciechowski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Umiejętności programowania. Podstawowa wiedza matematyczna ze szkoły średniej. Znajomość podstawowych algorytmów i mechanizmów sztucznej inteligencji.

Cel przedmiotu

Celem przedmiotu jest zapoznanie studentów z problemem sztucznej inteligencji w grach komputerowych, ze szczególnym uwzględnieniem gier odbywających się w czasie rzeczywistym. Studenci poznają rozwiązania pozwalające symulować inteligentne zachowania aktorów w grach na bazie profesjonalnych rozwiązań dostępnych m.in. w silniku gier Unreal Engine.

Przedmiotowe efekty uczenia się

Wiedza:

K2st_W2: ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną na temat problemu tworzenia sztucznej inteligencji w grach wideo

K2st_W3: ma zaawansowaną wiedzę szczegółową dotyczącą kluczowych zagadnień związanych z tworzeniem sztucznej inteligencji w grach w tym wykorzystaniem nowoczesnych silników gier i użyciem przygotowanych tam komponentów w zakresie wymagań tworzonego poziomu na potrzeby botów, oraz

samych mechanizmów budowania inteligentnego zachowania oraz sposobu prezentacji środowiska gry i poruszania się postaci

K2st_W5: ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia procesów sterujących sztuczną inteligencją botów w grach video

K2st_W6: zna zaawansowane narzędzia takie jak silnik Unreal Engine 5 stosowane przy rozwiązywaniu złożonych zadań inżynierskich w zakresie tworzenia inteligentnego zachowania agentów sterujących postaciami w grach komputerowych

Umiejętności:

K2st_U6: potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych, w zakresie wykorzystania sztucznej inteligencji w grach komputerowych

K2st_U8: potrafi dokonać krytycznej analizy oraz oceny sposobu funkcjonowania zachowania aktora sterowanego skryptami sztucznej inteligencji w grze

K2st_U9: potrafi ocenić przydatność metod i narzędzi takich jak BehaviourTree i Environment Query System w silniku Unreal Engine 5 służących budowie inteligentnego zachowania postaci sterowanej sztuczną inteligencją

K2st_U10: potrafi rozwiązać nietypowe zadanie dotyczące przetwarzania dane różnego typu, dokonywania ich syntezy do wiedzy i wniosków przydatnych do poprawy strategii aktorów sterowanych skryptami sztucznej inteligencji

K2st_U11: potrafi zaprojektować zaawansowane modele inteligentnych zachowań oraz zrealizować je z wykorzystaniem służące do tego celu komponentów silnika Unreal Engine

Kompetencje społeczne:

K2st_K1: rozumie, że odnośnie tematyki sztucznej inteligencji w grach wiedza i umiejętności szybko stają się przestarzałe, dostrzegając przy tym potrzebę ciągłego dokształcania oraz podnoszenia własnych kompetencji

K2st_K2: rozumie znaczenie wykorzystania najnowszej wiedzy związanej z informatyką i sztuczną inteligencją w rozwiązywaniu praktycznych problemów o kluczowym znaczeniu dla funkcjonowania jednostek i firm w takich obszarach zastosowań jak rozrywka

K2st_K4: ma świadomość potrzeby rozwijania dorobku zawodowego oraz przestrzegania zasad etyki zawodowej

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykłady: zaliczenie w formie pisemnej z pytaniami otwartymi przeprowadzane na ostatnim wykładzie. Każde pytanie będzie punktowane. Do zaliczenia trzeba uzyskać >50% punktów.

Laboratoria: Zajęcia będą pogrupowane na bloki tematyczne. Zaliczenie każdego z bloków wymaga przygotowania skryptów sztucznej inteligencji i opisu ich działania. Dodatkowo, dwa bloki kończą się turniejem, w którym rywalizują ze sobą przygotowane przez studentów skrypty. Do zaliczenia niezbędne jest oddanie wszystkich zadań.

Treści programowe

Tematyka przedmiotu dotyczy projektowania i implementacji zachowań aktorów z wykorzystaniem mechanizmów dostępnych w nowoczesnych silnikach gier na przykładzie silnika Unreal Engine 5. W ramach zajęć studenci odpowiedzialni są za programowanie bot-ów w specjalnie zaprojektowanych przykładowych środowiskach symulacyjnych.

Tematyka zajęć

Wprowadzenie do problematyki sztucznej inteligencji w grach komputerowych. Omówienie podstawowych rodzajów mechanizmów z uwzględnieniem kategorii gry. Cechy charakterystyczne i wymagania dotyczące sztucznej inteligencji w grach.

Omówienie środowiska eLua - platformy do nauki pisania skryptów sztucznej inteligencji (botów) w grach. Przedstawienie planszy na której toczy się rozgrywka, podstawowych instrukcji sterujących. Przedstawienie cech charakterystycznych języka LUA. Prezentacja sposobu implementacji takiego rozwiązania.

Problem poruszania się aktorów w zadanym środowisku i stosowane algorytmy wyszukiwania ścieżki.

Nowoczesne silniki gier - czym w ogóle jest silnik gier? Wprowadzenie do silnika gier Unreal Engine. Omówienie podstawowych modułów silnika i typów klas. Programowanie z wykorzystaniem blueprint'ów.

Przedstawienie komponentów sztucznej inteligencji w silniku. Prezentacja sposobu tworzenia postaci sterowanej sztuczną inteligencją. Inicjalizacja danych wejściowych oraz reagowanie na zmieniające się środowisko. Omówienie komponentów Blackboard i BehaviourTree.

Omówienie platformy AI Racer - stanowiącej środowisko oparte na silniku Unreal Engine umożliwiające tworzenie rozwiązań sterujących samochodami. Prezentacja podstawowych elementów platformy oraz sterowania samochodem. Budowa toru wyścigowego w środowisku.

Omówienie mechanizmów zapytań Environment Query System oferowanych w silniku Unreal Engine.

Przedstawienie komponentów systemu. Sposób integracji systemu zapytań z drzewem zachowań.

Turniej botów walczących evalUAtion - każdy ze studentów przygotowuje skrypty, które stanowią jego drużynę. Następnie skrypty te biorą udział w turnieju, rywalizując z pozostałymi drużynami.

Turniej botów w środowisku AI Racer - studenci przygotowują swoje skrypty sterujące samochodem, które następnie biorą udział w wyścigach na nieznanymi wcześniej torach wyścigowych.

Proceduralne generowanie treści np. mapy.

Omówienie problemów przy tworzeniu sztucznej inteligencji w grach w zależności od typu gry.

Metody dydaktyczne

Wykłady: prezentacje multimedialne dotyczące wybranych tematów połączone z prezentacją wybranych modułów omawianego oprogramowania.

Laboratoria: rozwiązywanie zadań prezentujących wybrane aspekty omawianego zagadnienia. Praca na komputerach w dedykowanych środowiskach.

Literatura

Podstawowa:

Mark DeLoura, Game Programming Gems (Game Programming Series), Charles River Media, 2000
lub

Mark DeLoura, tł. Rafał Jońca, Perełki programowania gier : vademecum profesjonalisty, T. 1, Helion, 2002

Mark DeLoura, Game Programming Gems 2 (Game Programming Series), Charles River Media, 2001
lub

Mark DeLoura, tł. Rafał Jońca, Perełki programowania gier: vademecum profesjonalisty, T. 2, Helion 2002

Dante Treglia, Game Programming Gems 3 (Game Programming Series), Charles River Media, 2002
lub

Dante Treglia, tł. Rafał Jońca, Perełki programowania gier: vademecum profesjonalisty, T. 3, Helion, 2003

Mike Dickheiser, Game Programming Gems 6 (Game Programming Series), Charles River Media, 2006
lub

Mike Dickheiser, tł. Mikołaj Szczepaniak, Perełki programowania gier: vademecum profesjonalisty, T. 6, Helion, 2008

Mat Buckland, Programming Game AI by Example, Jones & Bartlett Learning, 2004

Joanna Lee, Unreal Engine: nauka pisania gier dla kreatywnych, Helion 2017

Dokumentacja języka LUA <https://www.lua.org/docs.html>

Dokumentacja silnika Unreal Engine: <https://docs.unrealengine.com/5.0/en-US/>

Uzupełniająca:

Andrew Kirmse, Game Programming Gems 4 (Game Programming Series), Charles River Media, 2004

Kim Pallister, Game Programming Gems 5 (Game Programming Series), Charles River Media, 2005

Ian Millington, AI for Games, 3rd Edition, CRC Press, 2020

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	50	2,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	40	1,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	10	0,50