

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność		Programowanie gier komputerowych	
Semestr	IV	Program studiów,	2025/2026
Stopień studiów	I	dla którego obowiązuje sylabus	

Nazwa przedmiotu	Architektura i języki programowania gier komputerowych			
Kod przedmiotu	AIJPGK			
Łączna liczba godzin	18	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

Prowadzący zajęcia	
Forma prowadzonych zajęć	Laboratorium
Wymiar zajęć	18 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Wymagania wstępne	Podstawowa znajomość języków programowania, takich jak Java lub C++. Umiejętność obsługi komputera i środowisk programistycznych. Znajomość podstawowych algorytmów i struktur danych oraz podstaw grafiki komputerowej.
Założenia i cele przedmiotu	Przedmiot wprowadza studentów w podstawowe koncepcje architektury oprogramowania gier komputerowych, ze szczególnym uwzględnieniem struktur warstwowych i modułowych. Omawiane są kryteria wyboru języka programowania (C++, Python, C#), integracja logiki gry z silnikami fizyki i renderowania, zarządzanie zasobami oraz podstawowe metody testowania i debugowania. Celem jest wykształcenie umiejętności tworzenia stabilnej i skalowalnej architektury gry oraz efektywnego wykorzystania dostępnych technologii.
Metody dydaktyczne	<ol style="list-style-type: none"> 1. Prezentacje multimedialne. 2. Pokazy przykładowych rozwiązań problemów. 3. Rozwiązywanie zadań praktycznych.

Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)		Odniesienie do efektów dla kierunku	Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji
WIEDZA	W01.Podstawowe zasady projektowania architektury		

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

– absolwent zna i rozumie:	<p>gier, z uwzględnieniem warstw oprogramowania i modularnego podejścia.</p> <p>W02. Kryteria doboru języka programowania do realizacji zadań w projekcie gry oraz różnice między C++, Python i C#.</p> <p>W03. Zasady integracji logiki gry z silnikami fizyki i renderowania oraz modularnego podziału funkcjonalności.</p> <p>W04. Podstawy zarządzania zasobami w grach, w tym pamięcią, danymi multimedialnymi oraz metodami optymalizacji.</p> <p>W05. Metody i narzędzia testowania oraz debugowania gier, zrozumienie znaczenia procesu walidacji oprogramowania.</p>	<p>K_W02 K_W06 K_W07 K_W20 K_W21</p>	<p>P6S_WG P6S_WG_INŻ</p>
UMIEJĘTNOŚCI – absolwent potrafi:	<p>U01. Dokonać świadomego wyboru języka programowania i narzędzi odpowiednich do wymagań projektu gry.</p> <p>U02. Zaprojektować architekturę gry w sposób modułowy i zintegrować różne jej elementy (logika, renderowanie, fizyka).</p> <p>U03. Zaimplementować podstawowe algorytmy logiki gry oraz proste elementy silnika fizyki i renderowania.</p> <p>U04. Zarządzać zasobami gry w celu zapewnienia efektywnego wykorzystania pamięci i plików.</p> <p>U05. Zaplanować i przeprowadzić testy funkcjonalne oraz wykorzystać narzędzia debugujące do identyfikacji i usuwania błędów.</p>	<p>K_U01 K_U02 K_U03 K_U06 K_U08 K_U09 K_U15</p>	<p>P6S_UW P6S_UW_INŻ P6S_UO P6S_KK P6S_UK</p>
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	<p>K01. Pracy w zespole, przyjmując w nim różne role.</p> <p>K02. Krytycznej oceny możliwości urządzeń, oprogramowania i systemów dostępnych na rynku IT.</p> <p>K03. Ciągłego samokształcenia się w celu dostosowywania się do dynamicznie zmieniających się technologii.</p>	<p>K_K04 K_K05 K_K06</p>	<p>P6S_UO P6S_KR P6S_KK</p>

Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – laboratorium		
1	Wprowadzenie do architektury gier. Silniki i warstwy oprogramowania.	2
2	Wybór języka programowania do projektu gry: porównanie C++, Python, C#.	2
3	Projektowanie modułowe. Logika gry, silniki fizyki, renderowanie.	8

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

4	Zarządzanie zasobami w grach: pamięć, pliki, grafiki.	2
5	Podstawy testowania i debugowania gier: narzędzia i techniki. Zaliczenie.	4

Forma i warunki zaliczenia przedmiotu	Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Ocena projektów i częściowych prezentacji.	W01-W05, U01-U05, K01-K03

Literatura podstawowa	<ol style="list-style-type: none"> 1. R. Nystrom, <i>Programowanie gier. Wzorce</i>, PWN, Warszawa 2020. 2. A. Roman, <i>Testowanie i jakość oprogramowania. Modele, techniki, narzędzia</i>, Wydawnictwo Naukowe PWN, Warszawa 2015.
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. J. Gregory, <i>Game Engine Architecture</i>, PWN, Warszawa 2019. 2. E. Adams, <i>Projektowanie gier</i>, Helion 2010. 3. K. Hawkins, D. Astle, <i>OpenGL. Programowanie gier</i>, Helion 2003. 4. J. Matulewski, T. Dziubak, M. Sylwestrzak, R. Płoszajczak, <i>Grafika. Fizyka. Metody numeryczne</i>, PWN 2010. 5. B. M. T. de Sousa, <i>Programowanie gier. Kompendium</i>, Helion 2003.

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	18
Przygotowanie się do zajęć	9
Studiowanie literatury	9
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	22
Przygotowanie się do egzaminu / zaliczenia	-
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	60
Liczba punktów ECTS	2