

SYLABUS ZAJ /GRUPY ZAJ

Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	Grafika 3D i programowanie kart graficznych				
Course / group of courses:	3D Graphics and Graphics Cards Programming				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105986	Kod Erasmus:			
Punkty ECTS:	5	Rodzaj zaj :		fakultatywny	
Rok studiów:	2	Semestr:		3	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
2	3	LO	30	Zaliczenie z ocen	3
		W	30	Zaliczenie z ocen	2
Razem			60		5
Koordinator:	J drzej Byrski				
Prowadz cy zaj cia:					
J zyk wykładowy:	semestr: 3 - j zyk polski				

Obja nienia:

Rodzaj zaj : obowi zkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wiczenia audytoryjne, L - lektorat, S – seminarium/ zaj cia seminaryjne, P - wiczenia praktyczne (w tym zaj cia wf), M - wiczenia specjalistyczne (medyczne/ kliniczne), LO – wiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zaj cia z technologii informacyjnych, P – wiczenia projektowe, ZT – zaj cia terenowe, T - wiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR – praktyka, PR - praktyka zawodowa

Dane merytoryczne

Wymagania wst pne:			
Znajomo teoretyczna oraz praktyczna j zyka C oraz C++, zaliczenie pozytywne kursów: Programowanie w C, oraz Programowanie w C++.			
Szczegółowe efekty uczenia si			
Lp.	Student, który zaliczył zaj cia zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia si
1	Posiada wiedz w zakresie funkcjonowania biblioteki graficznej OpenGL.	IN1_W01, IN1_W04	wykonanie zadania, wypowied ustna
2	Zna architektury oraz zasady funkcjonowania sprz towych trójwymiarowych akceleratorów graficznych.	IN1_W03	kolokwium, wypowied ustna
3	Zna technologie i rozwi zania umo liwiaj ce wykorzystanie wielordzeniowych procesorów graficznych do oblicze in ynierskich.	IN1_W03, IN1_W01	wypowied ustna
4	Zna zagadnienia zwi zane z szybkim renderowaniem scen trójwymiarowych przy u yciu popularnych bibliotek graficznych.	IN1_W03, IN1_W04	wykonanie zadania, wypowied ustna

5	Zna zaawansowane zagadnienia związane z wykorzystaniem oraz programowaniem biblioteki CUDA.	IN1_W03, IN1_W04	wykonanie zadania, wypowiedź ustna
6	Zna architektury oraz zasady funkcjonowania sprzętowych trójwymiarowych akceleratorów graficznych.	IN1_W03, IN1_W11	wykonanie zadania, wypowiedź ustna
7	Wie, w którą stronę rozwija się nowoczesna grafika komputerowa na polu procesorów i obliczeń.	IN1_W11	kolokwium, wypowiedź ustna
8	Umie wykorzystać cechy programowalnych potoków przetwarzania grafiki przy tworzeniu oprogramowania dla akceleratorów graficznych.	IN1_U03	wykonanie zadania, wypowiedź ustna
9	Umie tworzyć równoległy algorytm rozwiązujący dany problem z wykorzystaniem koprocesora GPU.	IN1_U03	wykonanie zadania, wypowiedź ustna
10	Potrafi tworzyć oprogramowanie wykorzystujące bibliotekę OpenGL w procesach wytwarzania grafiki.	IN1_U03	kolokwium, wypowiedź ustna
11	Potrafi wykorzystać zaawansowane i niskopoziomowe funkcje platformy CUDA.	IN1_U03	kolokwium, wypowiedź ustna
12	Potrafi tworzyć zaawansowane oprogramowanie z wykorzystaniem platformy CUDA.	IN1_U03, IN1_U12	kolokwium, wypowiedź ustna
13	ma wiadomo, że ono ci relacji technologii i świata społecznego i znaczenia interdyscyplinarnej wiedzy przy tworzeniu i wdrażaniu rozwiązań technologicznych	IN1_U10	kolokwium, wypowiedź ustna
14	Potrafi samodzielnie rozwiązywać one problemy występujące podczas wytwarzania systemów opartych o trójwymiarowe biblioteki graficzne.	IN1_U12, IN1_U13	kolokwium, wypowiedź ustna

Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)

metody eksponujące (pokaz, prezentacja, objaśnienie (wyjaśnienie, omówienie).), metody podające (Wykład tradycyjny (informacyjny) z wykorzystaniem prezentacji (PP) i demonstracji przykładów, Wykład problemowy (obejmuje kompletny proces rozwiązania problemu od jego postawienia, po weryfikację rozwiązania).), metody praktyczne (ćwiczenia laboratoryjne.)

Kryteria oceny i weryfikacji efektów uczenia się

wiedza:

- ocena kolokwium (ocena kolokwium)
- ocena wykonania zadania (Wykonanie zadania)
- ocena wypowiedzi ustnej (Ocena wypowiedzi)

umiejętności:

- ocena kolokwium (ocena kolokwium)
- ocena wykonania zadania (Wykonanie zadania)
- ocena wypowiedzi ustnej (Ocena wypowiedzi)

Warunki zaliczenia

Zaliczenie laboratorium z oceną na podstawie kolokwium oraz zrealizowanych ćwiczeń laboratoryjnych oraz odpowiedzi ustnej, oceny wystawiane są zgodnie z aktualnym regulaminem studiów PWSZ w Tarnowie.

Treści programowe (opis skrócony)

Zapoznanie z podstawami grafiki 3D.
Zapoznanie z architekturą procesorów oraz kart graficznych.
Zapoznanie z architekturą oraz programowaniem CUDA.
Zapoznanie z podstawami programowania biblioteki OpenGL.

Content of the study programme (short version)

Introduction to the 3D graphics.
Introduction to architecture of processors and graphics cards.
Introduction to CUDA architecture and programming.
Introduction to the basics of OpenGL programming.

Treści programowe

	Liczba godzin
Semestr: 3	
Forma zajęć: wykład	

W ramach zajęć student zapozna się z zagadnieniami związanymi z renderowaniem trójwymiarowym przy użyciu biblioteki graficznej OpenGL, szybkim renderowaniu potoków wielokątów w sprzętowych akceleratorach 3D, przedstawione będą układy odwzorowania przestrzeni 3D, omówione będzie rysowanie figur prostych (prymitywów) i wyświetlanie siatek wielokątów, reprezentacja brył trójwymiarowych, wprowadzenie do OpenGL oraz tworzenie materiałów graficznych dla OpenGL. Omówione zostaną konwencje nazewnictwa funkcji i stałych bibliotecznych w OpenGL, typy zmiennych, obsługa błędów, podstawowe czynności konfiguracyjne, tworzenie aplikacji OpenGL, rysowanie w OpenGL, definiowanie obiektów rysowanych w OpenGL, definiowanie wierzchołków w układzie współrzędnych, definiowanie parametrów wyświetlania figur, przegląd funkcji bibliotecznych rysujących standardowe figury OpenGL oraz przekształcenia w przestrzeni trójwymiarowej OpenGL.

W części poświęconej programowaniu procesorów graficznych omówione zostaną obliczenia w grafice komputerowej (GPU) versus obliczenia w procesorze centralnym (CPU) i wpływ tej różnicy na rozwój dróg rozwojowych GPU i CPU, architektura stacji roboczych dla potrzeb grafiki komputerowej, architektura karty graficznej, koncepcja architektury dla potrzeb CUDA. Wprowadzenie teoretyczne do technologii CUDA, podstawowe oraz zaawansowane programowanie aplikacji korzystających z technologii CUDA, wraz z omówieniem zagadnień optymalizacji, podstawowe wiadomości o konstrukcji sprzętowych akceleratorów graficznych. Sprzęt renderujący i jego możliwości. Technologie programowania aplikacji z bezpośrednim wykorzystaniem GPU, technologie tworzenia kerneli w praktyce. Uruchomienie kerneli w aplikacjach bazujących na CUDA. Profile wymiany danych. Rozszerzenia biblioteki CUDA. Język nVidia CG i jego składnia. Dostępne funkcje i profile wymiany danych. Importowanie, kompilacja i uruchomienie programów.

Platforma CUDA (Compute Unified Device Architecture): Konfigurowanie platformy CUDA, wersje CUDA API i ich funkcjonalność, przykłady problemów obliczeniowych i ich rozwiązywanie. Programowanie niskopoziomowe z użyciem CUDA.

30

Forma zajęć : **wiczenia laboratoryjne**

W ramach zajęć student zapozna się z zagadnieniami związanymi z renderowaniem trójwymiarowym przy użyciu biblioteki graficznej OpenGL, szybkim renderowaniu potoków wielokątów w sprzętowych akceleratorach 3D, przedstawione będą układy odwzorowania przestrzeni 3D, omówione będzie rysowanie figur prostych (prymitywów) i wyświetlanie siatek wielokątów, reprezentacja brył trójwymiarowych, wprowadzenie do OpenGL oraz tworzenie materiałów graficznych dla OpenGL. Omówione zostaną konwencje nazewnictwa funkcji i stałych bibliotecznych w OpenGL, typy zmiennych, obsługa błędów, podstawowe czynności konfiguracyjne, tworzenie aplikacji OpenGL, rysowanie w OpenGL, definiowanie obiektów rysowanych w OpenGL, definiowanie wierzchołków w układzie współrzędnych, definiowanie parametrów wyświetlania figur, przegląd funkcji bibliotecznych rysujących standardowe figury OpenGL oraz przekształcenia w przestrzeni trójwymiarowej OpenGL.

W części poświęconej programowaniu procesorów graficznych omówione zostaną obliczenia w grafice komputerowej (GPU) versus obliczenia w procesorze centralnym (CPU) i wpływ tej różnicy na rozwój dróg rozwojowych GPU i CPU, architektura stacji roboczych dla potrzeb grafiki komputerowej, architektura karty graficznej, koncepcja architektury dla potrzeb CUDA. Wprowadzenie teoretyczne do technologii CUDA, podstawowe oraz zaawansowane programowanie aplikacji korzystających z technologii CUDA, wraz z omówieniem zagadnień optymalizacji, podstawowe wiadomości o konstrukcji sprzętowych akceleratorów graficznych. Sprzęt renderujący i jego możliwości. Technologie programowania aplikacji z bezpośrednim wykorzystaniem GPU, technologie tworzenia kerneli w praktyce. Uruchomienie kerneli w aplikacjach bazujących na CUDA. Profile wymiany danych. Rozszerzenia biblioteki CUDA. Język nVidia CG i jego składnia. Dostępne funkcje i profile wymiany danych. Importowanie, kompilacja i uruchomienie programów.

Platforma CUDA (Compute Unified Device Architecture): Konfigurowanie platformy CUDA, wersje CUDA API i ich funkcjonalność, przykłady problemów obliczeniowych i ich rozwiązywanie. Programowanie niskopoziomowe z użyciem CUDA.

30

Literatura

Podstawowa

Bharatkumar Sharma, Jack Han, CUDA Cookbook Effective recipes for parallel programming on GPU

Janusz Ganczarski, OpenGL. Podstawy programowania grafiki 3D
Jason Sanders, Edward Kandrot, CUDA w przykładach. Wprowadzenie do ogólnego programowania procesorów GPU
Kevin Hawkins, Dave Astle., OpenGL. Programowanie gier
Muhammad Mobeen Movania, OpenGL. Receptury dla programisty
Nicholas Wilt, The Cuda Handbook: A Comprehensive Guide to GPU Programming,
Richard S. Wright jr, Graham Sellers,, OpenGL. Księga eksperta
Tolga Soyata, GPU Parallel Program Development Using CUDA
Dodatkowa

Dane jako ciowe

Przyporzkowanie zaj /grup zaj do dyscypliny naukowej/artystycznej		informatyka techniczna i telekomunikacja	
Sposób okre lenia liczby punktów ECTS			
Forma nakładu pracy studenta (udział w zaj ciach, aktywno , przygotowanie sprawozdania, itp.)		Obci enia studenta [w godz.]	
Udział w zaj ciach		60	
Konsultacje z prowadz cym		2	
Udział w egzaminie		0	
Bezpo redni kontakt z nauczycielem - inne		0	
Przygotowanie do laboratorium, wicze , zaj		30	
Przygotowanie do kolokwiiów i egzaminu		10	
Indywidualna praca własna studenta z literatur , wykładami itp.		23	
Inne		0	
Sumaryczne obci enie prac studenta		125	
Liczba punktów ECTS			
Liczba punktów ECTS		5	
Zaj cia wymagaj ce bezpo redniego udziału nauczyciela akademickiego		L. godzin	ECTS
		62	2,5
Zaj cia o charakterze praktycznym		L. godzin	ECTS
		85	3,4

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zaj wymagaj cych bezpo redniego udziału nauczyciela akademickiego i o charakterze praktycznym mo e si ró ni od ł cznej liczby punktów ECTS dla zaj /grup zaj .