

Nazwa przedmiotu **Programowanie procesorów graficznych**

Kierunek studiów **Informatyka**

Poziom studiów **studia drugiego stopnia, stacjonarne**

Jednostka prowadząca WEEIA, Katedra Mikroelektroniki i Technik Informatycznych

Osoby proponujące treści kształcenia (Kierownik przedmiotu) **dr inż. Mariusz Zubert**, mgr inż. Adam Piotrowski

Proponowany przedmiot, formy zajęć, liczba godzin w semestrze	Semestr	w	ć	l	p	s	Liczba punktów
---	---------	---	---	---	---	---	----------------

	III	15		30			2
--	-----	----	--	----	--	--	---

Łączna liczba godzin proponowana do realizacji	45						2
--	----	--	--	--	--	--	---

Łączna liczba godzin zapisana w standardach

Wymagane umiejętności (wykaz przedmiotów z siatki godzin) Matematyczne metody w grafice komputerowej

Nazwa przedmiotu: Programowanie procesorów graficznych

Treści kształcenia proponowane do realizacji

WYKŁAD:

1. Techniki programowania procesorów graficznych
2. Języki wysokiego poziomu przeznaczone do programowania procesorów graficznych
 - 2.1. Język Sh
 - 2.2. Rozszerzenie języka Python – PyGPU
 - 2.3. Język Cg
3. Komunikacja bibliotek graficznych OpenGL i DirectX ze sprzętem
4. Podstawowe cechy shaderów w wersji 3
5. Wykorzystanie języka HLSL (High Level Shading Language)
6. Wybrane zaawansowane techniki programowania procesorów graficznych
 - 6.1. Wykorzystanie tablic LUT podczas transformacji kolorów
 - 6.2. Efektywne programowanie strumieni
 - 6.3. Optymalizacja obliczeń
 - 6.4. Zaawansowane techniki modelowania oświetlenia
 - 6.5. Efekty specjalne
7. Wykorzystanie procesorów graficznych do obliczeń niezwiązanych z przetwarzaniem grafiki
8. Niskopoziomowe programowanie procesorów graficznych
9. Emulatory konsol gier video dla komputerów PC
10. Podstawowe techniki programowania konsoli gier video
 - 10.1. Wykorzystanie techniki podwójnego buforowania
 - 10.2. Rendering obiektów 3D

LABORATORIUM:

1. Implementacja aplikacji korzystającej z niskopoziomowych technik dostępu do procesora graficznego.
2. Implementacja aplikacji wykorzystującej zaawansowane techniki renderingu

- obiektów
3. Implementacja aplikacji wykorzystującej procesor graficzny do obliczeń niezwiązanych z przetwarzaniem grafiki.
 4. Implementacja aplikacji działającej w środowisku konsoli gier video.

<i>Treści kształcenia w standardach</i>	brak
<i>Oczekiwane efekty kształcenia</i>	Praktyczna umiejętność programowania procesorów graficznych w języku C++. Umiejętność stosowania technik optymalizacji obliczeń, oraz wykorzystania możliwości GPU do obliczeń niezwiązanych z przetwarzaniem grafiki. Umiejętność programowania prostych aplikacji dla konsol gier video.
<i>Kwalifikacje i specjalność naukowa prowadzącego zajęcia</i>	Specjalność naukowa: informatyka.. Możliwość prowadzenia zajęć w języku angielskim.
<i>Wymagane wyposażenie techniczne - wykład</i>	Rzutnik multimedialny, dostęp do kompilatora języka C++ i narzędzi wspomagających uruchamianie programów (debugger).
<i>Wymagane wyposażenie techniczne (forma prowadzenia zajęć) - laboratorium</i>	Pracownia komputerowa z możliwością indywidualnej pracy studentów, wyposażona w kompilator języka C++, debugger oraz odpowiednie biblioteki graficzne
<i>Forma zaliczenia zajęć</i>	40% kolokwium zaliczające na wykładzie, 60% ocena ćwiczeń wykonanych w laboratorium.
<i>Literatura podstawowa</i>	M. Pharr, R. Fernando "GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation"
<i>Literatura uzupełniająca</i>	P. Metzger, „Anatomia PC. Wydanie X”, 2006 T. Kopacz „Karty graficzne VGA i SVGA – Programowanie grafiki”

<i>Obciążenie studenta (zajęcia w uczelni + praca własna)</i>	45+15
<i>Uwagi:</i>	