

Karta zaj

Informacje ogólne			
Nazwa zaj : Grafika i komunikacja człowiek-komputer			
Nazwa uczelni: Wy sza Szkoła Zarz dzania i Bankowo ci w Krakowie			
Wydział: Wydział Nauk Stosowanych			
Kierunek studiów: Informatyka			
Poziom studiów: pierwszego stopnia			
Forma studiów: niestacjonarne, stacjonarne	Profil kształcenia: praktyczny	Zakres kształcenia:	
Rok/Semestr: 2/4	Status zaj : obowi zkowy	J zyki wykładowe: polski	
Studia niestacjonarne	Forma zaj	wykłady	wiczenia laboratoryjne
	Wymiar zaj (w godz.)	16	16
Studia stacjonarne	Forma zaj	wykłady	wiczenia laboratoryjne
	Wymiar zaj (w godz.)	30	30
Koordynator zaj	dr in . Witold Alda		
Prowadz cy	dr in . Witold Alda dr in . Darin Nikolow		
Cele kształcenia	<p>C1. Dwa główne cele to zapoznanie si z podstawowymi poj ciami, modelami i algorytmami grafiki komputerowej oraz nabycie umiej tno ci w programowaniu grafiki na poziomie j zyka proceduralnego/obiekowego z wykorzystaniem bibliotek graficznych. Szczególowe cele wymienione s w kolejnych punktach, w kolejno ci która nie odpowiada omawianiu tematów na wykładach.</p> <p>C2. Ogólne zapoznanie si ze sprz tem graficznym (kartami/procesorami graficznymi), jego charakterystyk i specyfik wykorzystania - w tym podstaw programowania.</p> <p>C3. Zapoznanie si z podstawami modelowania obiektów 3D, zasadami budowania sceny oraz rzutowaniem za pomoc kamery.</p> <p>C4. Zapoznanie si z podstawowymi modelami o wietlenia scen 3D, rodzajami ródeł wiatła oraz parametrami materiałowymi charakteryzuj cymi sposób odbicia wiatła od powierzchni obiektów geometrycznych.</p> <p>C5. Pogł bienie umiej tno ci w programowaniu JavaScript w zakresie potrzebnym do grafiki komputerowej. Zapoznanie si z graficzn bibliotek wysokopoziomow Three.js oraz bibliotekami pomocniczymi i nabycie praktycznych umiej tno ci w korzystaniu z nich.</p> <p>C6. Zapoznanie si z niskopoziomow bibliotek graficzn WebGL oraz j zykiem shaderów GLSL w stopniu elementarnym.</p> <p>C7. Zapoznanie si z poj ciami i koncepcj tekstur oraz praktycznymi sposobami ich wykorzystania w grafice komputerowej.</p> <p>C8. Zapoznanie si z matematycznymi podstawami transformacji geometrycznych, obejmuj cymi macierze transformacji i operacje macierzowo-wektorowe.</p> <p>C9. Zapoznanie si z podstawowymi matematycznymi reprezentacjami form geometrycznych: funkcjami parametrycznymi (zwłaszcza wielomianami Beziera), funkcjami w postaci uwikłanej i funkcjami fraktalnymi (do modelowania obiektów naturalnych).</p> <p>C10. Zapoznanie si programowaniem obsługi zdarze w aplikacjach interaktywnych. Dokonanie przegl du i zapoznanie si z ró nymi urz dzeniami do komunikacji człowiek-komputer.</p> <p>C11. Zapoznanie si z modelem cz stek w grafice komputerowej, jego przykładowymi implementacjami i zastosowaniami w grafice komputerowej.</p>		

Wymagania wstępne	Wymagana jest podstawowa znajomość programowania. Przydatna może być (choć niekonieczna) znajomość języka JavaScript. Przydatna jest również znajomość podstawowych pojęć algebry: wektorów i macierzy oraz operacji na nich.
-------------------	---

Efekty uczenia się			Odniesienie do efektów uczenia się dla kierunku	Odniesienie do charakterystyk PRK poziomu 6
Wiedza	EU1	Student zna i rozumie podstawowe pojęcia i elementy grafiki 2D i 3D, a także podstawy matematyczne tych pojęć. Jest ogólnie zorientowany w architekturze kart/procesorów graficznych i innego sprzętu używanego w grafice komputerowej.	K_W04	P6U_W P6S_WG
	EU2	Student zna narzędzia (biblioteki, pakiety programów, aplikacje) grafiki 3D, wie jakie są ich możliwości, jak i w jakich sytuacjach należy je użyć.	K_W05	P6U_W P6S_WG
Umiejętności	EU3	Student umie sprawnie zbudować interaktywną aplikację grafiki 3D w języku JavaScript, wykorzystując biblioteki Three.js i inne biblioteki pomocnicze.	K_U01 K_U05 K_U09	P6U_U P6S_UW P6S_UU P6S_UK
Kompetencje społeczne	EU4	Student rozumie i jest otwarty na wykorzystanie grafiki komputerowej w wielu dziedzinach działalności ludzkiej; umie ją w nich twórczo zastosować.	K_K02 K_K04	P6U_U P6S_KO

Treści programowe	
Wykład	
W1	Wprowadzenie do grafiki komputerowej. Klasyfikacja grafiki komputerowej. Scharakteryzowanie grafiki rastrowej i wektorowej, grafiki 2D i 3D. Analiza podstawowej postaci potoku graficznego nieprogramowanego i programowanego. Podstawowe pojęcia: układ współrzędnych 3D, wierzchołki, krawędzie i ściany, materiały, tekstury, światło, transformacje i rzutowanie, kamery, użycie macierzy, Wprowadzenie do programowania grafiki w WebGL. Przypomnienie wiadomości o HTML5 i JavaScript
W2	Wprowadzenie do WebGL. Charakterystyka i historia biblioteki WebGL. Nawigowanie do bibliotek OpenGL i OpenGL ES i Vulkan. Omówienie konstrukcji biblioteki, podstawowych funkcji, stałych i zmiennych systemowych. Przedstawienie struktury typowego programu. Wspomnienie o ewolucji WebGL, krótkie porównanie wersji 1 i 2.
W3	Dwa style programowania w WebGL. Programowanie niskopoziomowe z użyciem buforów, transformacji macierzowych i programów cieniujących. Programowanie wysokopoziomowe z użyciem biblioteki Three.js i pokrewnych, oparte na podejściu obiektowym. Porównanie zalet i wad obu podejść oraz zilustrowanie ich na przykładach. Omówienie elementów biblioteki three.js i konstruowania programu: tworzenie renderera, tworzenie sceny i umieszczanie na niej obiektów, wprowadzenie głównej pętli programu, zastosowanie oświetlenia. Interakcja, grafika interaktywna i tworzenie programów sterowanych zdarzeniami. Demonstracja typowych przykładów
W4	Postrzeganie barwne. Modele oświetlenia. Krótkie omówienie teorii barwy i podstawowych modeli barwnych. Ogólne dywagacje na temat złożoności i efektywności modelu oświetlenia. Omówienie podstawowego modelu lokalnego ADS. Wprowadzenie normalnych do obliczania oświetlenia. Omówienie modeli Phong, Blinn i interpolacji oświetlenia według Gourauda. Modele oświetlenia i rodzaje ról światła dostępne w three.js. Omówienie parametrów materiałowych powierzchni. Informacja o modelach globalnego oświetlenia.
W5	Mapowanie tekstur. Podstawy. Podstawowe pojęcia dotyczące tekstur i ich parametrów. Wczytywanie tekstury z pliku. Tworzenie listy tekstur, mapowanie tekstur na obiekty geometryczne. Metody w Three.js do mapowania tekstur. Zaawansowane mapowanie tekstur. Multiteksturowanie, ładowanie tekstur z oświetleniem, mieszanie oświetlenia, omówienie techniki mipmappingu. Użycie tekstur do mapowania obrazu otoczenia (sphere mapping, cube mapping). Techniki optycznego uwypuklenia nierówności tekstury (bump mapping, normal mapping, parallax mapping). Tekstury proceduralne. Informacja o teksturach 3D.

W6	Transformacje geometryczne. Transformacje 3D. Omówienie przekształceń afinicznych. Reprezentacja przekształceń przez macierz transformacji. Używanie funkcji transformacji i bezpośrednie operowanie na macierzach. Składanie transformacji. Interpretacja transformacji w lokalnym i globalnym układzie współrzędnych. Ruch obiektów, a ruch kamery. Rzutowanie: rodzaje, parametry i macierze rzutowania. Realizacja transformacji w three.js
W7	Nawigowanie po scenie przy pomocy klawiatury i myszki, przy wykorzystaniu dostępnych bibliotek w JavaScript. Porównanie technik znanych z gier komputerowych i programów do modelowania obiektów: First Person Controller, Orbit Controller, Trackball Controller, chwytanie i przesuwanie obiektów
W8	Człotki w grafice komputerowej. Obiekty typu sprite. Modelowanie kinematyki i dynamiki chmury człotek. Zastosowania modeli człotek w grafice komputerowej. Modelowanie obiektów zbudowanych z polczonych człotek: tkanin i obiektów elastycznych 3D.
W9	Tworzenie i edycja obiektów geometrycznych oraz ich przykładowa implementacja w bibliotece Three.js. Operacje logiczne na obiektach geometrycznych. Operacje Extrude i Tube. Modelowanie krzywych i powierzchni parametrycznych za pomocą wielomianów Beziery i funkcji NURBS. Modelowanie za pomocą powierzchni uwikłanych. Graficzne modelowanie zjawisk i procesów naturalnych w grafice komputerowej. Omówienie elementów geometrii fraktalnej.
W10	Architektura i programowanie procesorów graficznych. Przedstawienie logicznej struktury procesorów graficznych, na wybranym przykładzie. Omówienie współpracy CPU i GPU. Przedstawienie podstaw języka programów cieniujących (shaderów) GLSL i zasad włączania go do programu WebGL/Three.js.
W11	Przełd nowoczesnych narzędzi komunikacji człowieka z komputerem. Omówienie podstaw wirtualnej i rozszerzonej rzeczywistości. Przedstawienie perspektyw dla grafiki komputerowej.

Laboratorium	
L1	Laboratorium wprowadzające. Zapoznanie się z procedurą uruchamiania i debuggowania programów.
L2	Uruchomienie programu z prostą sceną 3D. Przypomnienie. Wykorzystanie gotowych przykładów do uruchomienia. Powtórzenie i utrwalenie wiadomości o JavaScript.
L3	Przełd bibliotek pomocniczych używanych wraz z Three.js. Demonstracja na przykładach jakie mają możliwości i jak ich używać (miedzy innymi omówienie działania bibliotek jquery.js, stats.js, dat.gui.js).
L4	Uruchomienie i modyfikacja podstawowych programów: ustawianie kamery i jej parametrów, umieszczanie gotowych obiektów Three.js na scenie, dodawanie i usuwanie obiektów.
L5	ćwiczenia w wykonaniu transformacji geometrycznych obiektów: skalowaniu, przesuwaniu i obrotach
L6	Uruchamianie przykładów w nawigowaniu po scenie: przelot nad sceną, zaznaczanie i przesuwanie obiektów
L7	Modelowanie oświetlenia. Wykorzystanie możliwości jakie daje biblioteka Three.js w oświetleniu lokalnym. Wprowadzenie w przykładach klasycznych modeli ródeł światła (punktowe, kierunkowe, otoczenia, reflektor) oraz rozszerze dostarczonych przez Three.js (area light, hemisphere light). Powiązanie ródeł światła z parametrami materiałowymi powierzchni w celu uzyskania po danych efektów.
L8	Nakładanie tekstur. Omówienie podstawowych funkcji obsługi tekstur. Omówienie funkcji do wczytywania obrazów rastrowych jako tekstur. Uzupełnienie wcześniejszych przykładów o nowe tekstury.
L9	Nakładanie tekstur typu Sphere mapping i Skybox oraz efekty specjalne. Użycie techniki Sphere mapping do wprowadzenia obiektów lustrzanych. Tworzenie otoczenia sceny za pomocą techniki Skybox. Rozszerzenie przykładów o nowe efekty. Użycie tablic tekstur do efektów animacji.
L10	ćwiczenia w importowaniu zewnętrznych modeli 3D oraz ich oświetleniu i teksturuwaniu, a także animacji w przypadku formatu fbx.
L11	Modelowanie zjawisk fizycznych za pomocą człotek. W ramach ćwicze demonstrowane s przykłady niegu, deszczu i fajerwerków.
L12	ćwiczenia uwzgl dniające aspekty fizyczne obiektów (kolizje, bezwładność, tarcie) przy wykorzystaniu możliwości biblioteki Physijs.
L13	Realizacja programów z shaderami. Wprowadzenie shaderów do poprzednich przykładów w celu wzbogacenia oświetlenia i teksturuwania.

Ocena studenta			
Metody/Narzędzia dydaktyczne	N1	prezentacja multimedialna	wykład
	N2	wykonanie ćwicze laboratoryjnych	laboratorium

	N3	materiały dydaktyczne dostępne w SAKE	wykład laboratorium
Sposoby oceny/metody weryfikacji uczenia się	Ocena formująca		
	F1	średnia ocen uzyskanych podczas zajęć	laboratorium
	F2	Ocena z kolokwium/kolokwiów	laboratorium
	F3	Test komputerowy	wykład
	Ocena podsumująca		
	P1	Ocena z egzaminu/zaliczenia	wykład
	P2	Ocena z ćwiczeń laboratoryjnych	laboratorium

Kryteria oceny				
	EU1	EU2	EU3	EU4
Na ocenę 3	51%	51%	51%	51%
Na ocenę 3,5	62%	62%	62%	62%
Na ocenę 4	74%	74%	74%	74%
Na ocenę 4,5	86%	86%	86%	86%
Na ocenę 5	95%	95%	95%	95%

Literatura	
Literatura podstawowa	1. Matusiewicz Zofia: Matematyka dla grafików komputerowych. Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania, Rzeszów 2008 2. Dirksen Jos: Learning Three.js - the JavaScript 3D Library for WebGL. Packt Publishing, Birmingham-Mumbai 2015
Literatura uzupełniająca	1. Dirksen Jos: Three.js Essentials. Packt Publishing, Birmingham 2014 2. Angel Edward, Shreiner Dave: Interactive Computer Graphics. Pearson, Boston 2015

Nakład pracy studenta		
	Studia niestacjonarne	Studia stacjonarne
Godziny kontaktowe z nauczycielem akademickim lub inną osobą prowadzącą zajęcia (wykłady, ćwiczenia, laboratoria, konwersatoria)	32	60
Przygotowanie do zajęć, w tym studiowanie zalecanej literatury podstawowej i uzupełniającej	33	35
Przygotowanie projektu	16	10
Przygotowanie się do egzaminu / zaliczenia	19	20
Inne (np. esej, prezentacja, referat, koreferat, sprawozdanie z wykonanych zadań)	0	0
Łączny nakład pracy studenta w godz.	100	125
Liczba punktów ECTS	4	4

Macierz realizacji zajęć					
Efekty uczenia się	Odniesienie danego efektu do kierunkowych efektów uczenia się	Cele kształcenia	Treści programowe	Metody/Narzędzia dydaktyczne	Sposoby oceny
EU1	K_W04	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11	W1, W4, W5, W6, W7, W8, W9, W10, W11, L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13	N1, N2, N3	F1, F2, F3, P1, P2

EU2	K_W05	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11	W2, W3, W4, W5, W6, W7, W8, L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13	N1, N2, N3	F1, F2, F3, P1, P2
EU3	K_U01, K_U05, K_U09	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11	W2, W3, W4, L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13	N1, N2, N3	F1, F2, F3, P1, P2
EU4	K_K02, K_K04	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11	W5, W8, W10, W11, L1	N1, N2, N3	F1, F2, F3, P1, P2