

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie gier		Kod 1010515321010511655
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 2
Ścieżka obieralności/specjalność Zaawansowane technologie internetowe	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: 18 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Paweł Wojciechowski email: Pawel.Wojciechowski@cs.put.poznan.pl tel. 61 6653031 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu programowania, grafiki trójwymiarowej, sieci komputerowych, sztucznej inteligencji oraz matematyki w zakresie działań na wektorach i macierzach.
2	Umiejętności:	Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu algorytmiki, optymalizacji kodu programu, pracy z bibliotekami zewnętrznymi dla języka C/C++ oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
1. Przekazanie studentom podstawowej wiedzy o problemach spotykanych przy programowaniu gier video, w zakresie sposobu działania silników grafiki oraz fizyki zarówno dwu, jaki i trójwymiarowych, działania skryptów sztucznej inteligencji i niskopoziomowego programowania efektów wizualnych, technik animacji oraz modelowania trójwymiarowego.		
2. Rozwijanie u studentów umiejętności przygotowania modelu trójwymiarowego danego obiektu wraz z jego animacją, wykorzystania gotowych bibliotek graficznych i symulacji fizyki.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie: funkcjonalności silników grafiki dwu i trójwymiarowej, sposobu opisu świata dla skryptów sztucznej inteligencji w grach, potoku przetwarzania grafiki trójwymiarowej - [K2st_W2]		
2. ma szczegółową wiedzę w zakresie architektury i działania silników grafiki 3D zna podstawowe wymagania dotyczące symulacji fizyki i łączenia jej z wizualizacją - [K2st_W3]		
3. zna podstawowe techniki animacji modeli 3D, zna przykładowe narzędzie stosowane do modelowaniu obiektów trójwymiarowych - [K2st_W3]		
4. ma podstawową wiedzę o trendach w ewolucji trójwymiarowej grafiki komputerowej i kart graficznych - [K2st_W4]		
Umiejętności:		

1. potrafi pozyskiwać informacje z literatury, dokumentacji oraz forum dyskusyjnym dotyczącym silnika UnrealEngine 4 (w języku polskim i angielskim), integrować je, dokonywać ich interpretacji i stosować alternatywne rozwiązania do podanych na zajęciach - [K2st_U1]
2. potrafi przy formułowaniu i rozwiązywaniu zadań inżynierskich takich jak programowanie prostej gry, integrować wiedzę zarówno z różnych obszarów informatyki (np programowanie obiektowe, metody sztucznej inteligencji, programowanie sieciowe, przetwarzanie danych obiektów trójwymiarowych), jak i innych dyscyplin naukowych takich jak fizyka - [K2st_U5]
3. potrafi ocenić przydatność i możliwość wykorzystania poszczególnych wersji modułów silnika UnrealEngine 4 - [K2st_U6]
4. potrafi ocenić przydatność silników grafiki 3D wspomagających programistę w procesie implementacji gry, w tym dostrzec celowość ich stosowania - [K2st_U8]
5. zna i potrafi przyjmować różne role w procesie budowy produktu jakim jest gra komputerowa - [K2st_U15]
Kompetencje społeczne:
1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]

Sposoby sprawdzenia efektów kształcenia

Ocena formująca:

- a) w zakresie wykładów:
 - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,
- b) w zakresie laboratoriów:
 - na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
 - ocenę wiedzy w trakcie egzaminu końcowego
- b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
 - ocenę wiedzy i umiejętności związanych z tematyką w formie małego zadania projektowego w semestrze tj. przygotowanie walczących skryptów sztucznej inteligencji,
 - ocenę i obronę przez studenta projektu końcowego, dotyczącego implementacji prostej gry video

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za uwagi związane z udoskonaleniem materiałów dydaktycznych.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Problematyka sztucznej inteligencji w grach realizowana jest w oparciu o projekt evalUAtion przygotowany w ramach zajęć w latach ubiegłych. W ramach wykładu, przedstawione są podstawowe założenia środowiska, omówiony problem interakcji skryptu z otoczeniem i sposób opisu otoczenia widzianego z poziomu skryptu.

Wprowadzenie do silnika gier UnrealEngine. Budowanie zasobów gier dwuwymiarowych - przygotowanie obrazków i animacja duszków (sprite'ów) oraz zarządzanie zasobami. Interakcja obiektów gry ze środowiskiem. Omówienie rodzajów obiektów występujących w grze i ich właściwości. Wprowadzenie właściwości fizycznych wybranych obiektów w tym sposoby ograniczania symulacji do dwóch wymiarów.

Zasady modelowania 3D na przykładzie aplikacji Blender. Zapoznanie się z narzędziem, podstawowe własności modeli, metody teksturowania, rodzaje oświetlenia, mapowanie nierówności. Rodzaje animacji w grafice 3D na przykładzie programu Blender. Pojęcia klatek kluczowych, rodzaje interpolacji, animacja szkieletowa oraz deformacje siatki obiektów (ang. Shape Keys). Zasady eksportu modeli i animacji do silnika UnrealEngine.

Wykorzystanie modelu w procesie budowy gry 3D i importu standardowej postaci.

Właściwości silnika grafiki 3D Ogre3D, podstawowe przekształcenia obiektów, import modeli z programu Blender, animacje, system tworzenia terenu, sposoby realizacji otoczenia (SkyBox, SkyPlane). Wykorzystanie fizyki. Przedstawienie problemów i możliwości badanych rozwiązań.

Omówienie potoku grafiki 3D, ewolucja kart graficznych, języki ich programowania, wprowadzenie podstawowych pojęć: pixel shader, vertex shader, geometry shader, cieniowanie wierzchołków i pikseli, mapowanie nierówności.

Zajęcia laboratoryjne prowadzone są w formie piętnastu 3-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych 2-godzinną sesją instruktazową na początku semestru. Program laboratorium obejmuje wykorzystanie bibliotek zewnętrznych dla różnych języków programowania wykorzystywanych w programowaniu gier. Zajęcia laboratoryjne podzielone są na bloki tematyczne:

Sztuczna inteligencja, w ramach której studenci zapoznają się ze środowiskiem evalUAtion służącym do nauki pisania skryptów walczących w języku LUA. Każdy ze studentów przygotowuje drużynę składającą się z trzech postaci, które następnie biorą udział w turnieju. Następnie, w późniejszym terminie studenci realizują zadania związane z wykorzystaniem mechanizmów sztucznej inteligencji w silniku UE4.

Wprowadzenie do silnika UnrealEngine - wykorzystanie podstawowych narzędzi, blueprints na przykładzie implementacji prostej gry platformowej z wykorzystaniem modelu przygotowanego wraz z teksturami w środowisku blender.

Omówienie zagadnień związanych z modułem fizyki w środowisku UE4 - podstawowe mechanizmy kolizji, ruchu oraz zależności między obiektami. Obsługa podstawowych zdarzeń generowanych w trakcie zderzeń obiektów. Wprowadzenie do pojęcia materiału fizycznego.

Omówienie materiałów i wprowadzenie do systemu cząstek. Wykorzystanie elementów interfejsu użytkownika (GUI).		
Metody dydaktyczne:		
1. wykład: prezentacje multimedialna, wraz z demonstracją następującego oprogramowania: evaLUAtion, Blender, UnrealEngine4		
2. ćwiczenia laboratoryjne: rozwiązywanie zadań obejmujących wykorzystanie wybranych technologii, turniej walczących skryptów		
Literatura podstawowa:		
1. Perełki programowania gier t. 1, 2 i 3, DeLoura M., Helion, 2002		
2. Fizyka dla programistów gier, Bourg, D.M., O'Reilly; Associates, 2003		
3. Mathematics for 3D Game Programming and Computer Graphics, Third Edition, Lengyel, E., Course Technology PTR, 2011		
4. Programming Game AI by Example, Buckland, M., Jones Bartlett Publishers, 2004		
5. Game Coding Complete 4 edition, McShaffry, M., Graham, D., Course Technology PTR, 2012		
6. Język Cg. Programowanie grafiki w czasie rzeczywistym, Fernando, R., Kilgard, M.J., Helion, 2003		
7. Blender. Kompendium, Kuklo, K., Kolmaga, J., Helion, 2007		
8. Animacja komputerowa Algorytmy i techniki, Parent, R., PWN, 2011		
Literatura uzupełniająca:		
1. GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics, Fernando, R. (Series Editor), Addison Wesley Professional, 2004		
2. GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, Pharr, M., Fernando, R. (Series Editor), Addison Wesley Professional, 2005		
Bilans nakładu pracy przeciętnego studenta		
Czynność		Czas (godz.)
1. udział w zajęciach laboratoryjnych / ćwiczeniach		18
2. przygotowanie do zajęć laboratoryjnych, praca własna nad wybranymi zagadnieniami		10
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną)		2
4. napisanie programu zaliczeniowego, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)		25
5. napisanie skryptów sztucznej inteligencji		6
6. udział w wykładach		16
7. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron		10
8. przygotowanie modelu i jego animacji w programie Blender		8
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	95	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1
Zajęcia o charakterze praktycznym	77	3