

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Programowanie gier</b>		Kod <b>1010512311010511655</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 1</b>
Ścieżka obieralności/specjalność <b>Gry i technologie internetowe</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>15</b> Ćwiczenia: - Laboratoria: <b>45</b> Projekty/seminaria: -		Liczba punktów <b>4</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>  <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>4 100%</b>  <b>4 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>  dr inż. Paweł Wojciechowski email: Pawel.Wojciechowski@cs.put.poznan.pl tel. 61 6653031 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	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	<b>Umiejętności:</b>	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	<b>Kompetencje społeczne</b>	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.
<b>Cel przedmiotu:</b>		
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 oraz programowania kart graficznych.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
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]		
<b>Umiejętności:</b>		

<p>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]</p> <p>2. potrafi wykorzystać rozwiązywania zadań związanych z opracowaniem skryptów sztucznej inteligencji metody eksperymentalne i symulacyjne oferowane przez silnik UnrealEngine 4 i platformę evalUAtion - [K2st_U4]</p> <p>3. 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]</p> <p>4. potrafi ocenić przydatność i możliwość wykorzystania poszczególnych wersji modułów silnika UnrealEngine 4 - [K2st_U6]</p> <p>5. potrafi ocenić przydatność silników grafiki 3D wspomagających programistę w procesie implementacji gry, w tym dostrzec celowość ich stosowania - [K2st_U8]</p> <p>6. zna i potrafi przyjmować różne role w procesie budowy produktu jakim jest gra komputerowa - [K2st_U15]</p>
<b>Kompetencje społeczne:</b>
<p>1. rozumie, że w obszarze gier komputerowych wiedza i umiejętności bardzo szybko stają się przestarzałe, stąd istnieje potrzeba ciągłego uczenia się - [K2st_K1]</p>

Sposoby sprawdzenia efektów kształcenia
<p>Ocena formująca:</p> <p>a) w zakresie wykładów:</p> <ul style="list-style-type: none"><li>- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,</li></ul> <p>b) w zakresie laboratoriów:</p> <ul style="list-style-type: none"><li>- na podstawie oceny bieżącego postępu realizacji zadań,</li></ul> <p>Ocena podsumowująca:</p> <p>a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none"><li>- ocenę wiedzy i umiejętności zawartych w projekcie końcowym realizowanym indywidualnie w ramach laboratoriów</li></ul> <p>b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none"><li>- ocenę wiedzy i umiejętności związanych z tematyką w formie dwóch małych zadań projektowych w semestrze tj. przygotowanie walczących skryptów sztucznej inteligencji oraz animowanego modelu 3D,</li><li>- ocenę i obronę przez studenta projektu końcowego, dotyczącego implementacji prostej gry video</li></ul> <p>Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za uwagi związane z udoskonaleniem materiałów dydaktycznych.</p>
Treści programowe
<p>Program wykładu obejmuje następujące zagadnienia:</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Wykorzystanie modelu w procesie budowy gry 3D i importu standardowej postaci.</p> <p>Właściwości silnika grafiki 3D Unreal Engine, podstawowe przekształcenia obiektów, import modeli z programu Blender, animacje, system tworzenia terenu. Wykorzystanie fizyki. Przedstawienie problemów i możliwości badanych rozwiązań.</p> <p>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.</p> <p>Zajęcia laboratoryjne prowadzone są w formie piętnastu 3-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych 2-godzinną sesją instruktażową na początku semestru. Program laboratorium obejmuje wykorzystanie bibliotek zewnętrznych dla różnych języków programowania wykorzystywanych w programowaniu gier. Zajęcia laboratoryjne podzielone są na bloki tematyczne:</p> <p>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.</p> <p>Wprowadzenie do silnika UnrealEngine - wykorzystanie podstawowych narzędzi, blueprints na przykładzie implementacji prostej gry platformowej z wykorzystaniem modelu przygotowanego wraz z teksturami w środowisku blender.</p> <p>Omówienie zagadnień związanych z modulem fizyki w środowisku UE4 - podstawowe mechanizmy kolizji, ruchu oraz zależności między obiektami. Obsługa podstawowych zdarzeń generowanych w trakcie zderzeń obiektów. Wprowadzenie do</p>

<p>pojęcia materiału fizycznego. Omówienie materiałów i wprowadzenie do systemu cząstek. Wykorzystanie elementów interfejsu użytkownika (GUI). Wprowadzenie do wykorzystania języka C++ do implementacji gry.</p> <p>Metody dydaktyczne:</p> <ol style="list-style-type: none"> <li>1. wykład: prezentacje multimedialna, wraz z demonstracją następującego oprogramowania: evaLUAtion, Blender, UnrealEnigne4</li> <li>2. ćwiczenia laboratoryjne: rozwiązywanie zadań obejmujących wykorzystanie wybranych technologii, turniej walczących skryptów</li> </ol>		
<p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"> <li>1. Perełki programowania gier t. 1, 2 i 3, DeLoura M., Helion, 2002</li> <li>2. Fizyka dla programistów gier, Bourg, D.M., O'Reilly; Associates, 2003</li> <li>3. Mathematics for 3D Game Programming and Computer Graphics, Third Edition, Lengyel, E., Course Technology PTR, 2011</li> <li>4. Programming Game AI by Example, Buckland, M., Jones Bartlett Publishers, 2004</li> <li>5. Game Coding Complete 4 edition, McShaffry, M., Graham, D., Course Technology PTR, 2012</li> <li>6. Język Cg. Programowanie grafiki w czasie rzeczywistym, Fernando, R., Kilgard, M.J., Helion, 2003</li> <li>7. Blender. Kompendium, Kuklo, K., Kolmaga, J., Helion, 2007</li> <li>8. Animacja komputerowa Algorytmy i techniki, Parent, R., PWN, 2011</li> </ol>		
<p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"> <li>1. GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics, Fernando, R. (Series Editor), Addison Wesley Professional, 2004</li> <li>2. GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, Pharr, M., Fernando, R. (Series Editor), Addison Wesley Professional, 2005</li> </ol>		
<p><b>Bilans nakładu pracy przeciętnego studenta</b></p>		
<p><b>Czynność</b></p>		<p><b>Czas (godz.)</b></p>
<p>1. udział w zajęciach laboratoryjnych / ćwiczeniach</p>		<p>45</p>
<p>2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną)</p>		<p>2 16</p>
<p>3. napisanie programu zaliczeniowego, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)</p>		<p>8 15</p>
<p>4. napisanie skryptów sztucznej inteligencji</p>		<p>10</p>
<p>5. udział w wykładach</p>		<p>5</p>
<p>6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron</p>		<p>5</p>
<p>7. przygotowanie modelu i jego animacji w programie Blender</p>		<p></p>
<p><b>Obciążenie pracą studenta</b></p>		
<p><b>forma aktywności</b></p>	<p><b>godzin</b></p>	<p><b>ECTS</b></p>
<p>Łączny nakład pracy</p>	<p>101</p>	<p>4</p>
<p>Zajęcia wymagające bezpośredniego kontaktu z nauczycielem</p>	<p>68</p>	<p>3</p>
<p>Zajęcia o charakterze praktycznym</p>	<p>74</p>	<p>3</p>