

| <b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>  |  |  |
|--|--|--|
| Nazwa modułu/przedmiotu<br><b>Programowanie gier</b>   |  | Kod<br><b>1010512321010511655</b>  |
| Kierunek studiów<br><b>Informatyka</b>   | Profil kształcenia (ogólnoakademicki, praktyczny)<br><b>ogólnoakademicki</b> | Rok / Semestr<br><b>1 / 2</b>  |
| Ścieżka obieralności/specjalność<br><b>Internet Przedmiotów</b>  | Przedmiot oferowany w języku:<br><b>polski</b>                               | Kurs (obligatoryjny/obieralny)<br><b>obieralny</b>   |
| Stopień studiów:<br><b>II stopień</b>  | Forma studiów (stacjonarna/niestacjonarna)<br><b>stacjonarna</b>             |  |
| Godziny<br>Wykłady: <b>30</b> Ćwiczenia: - Laboratoria: <b>30</b> Projekty/seminaria: -  |  | Liczba punktów<br><b>4</b>   |
| Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny)<br><b>kierunkowy</b>  |  | (ogólnouczelniany, z innego kierunku)<br><b>z danego kierunku</b>  |
| Obszar(y) kształcenia i dziedzina(y) nauki i sztuki<br><b>nauki techniczne</b>   |  | Podział ECTS (liczba i %)<br><b>4 100%</b>   |
| <b>Odpowiedzialny za przedmiot / wykładowca:</b>   |  |  |
| <p>dr inż. Paweł Wojciechowski<br/>                     email: Pawel.Wojciechowski@cs.put.poznan.pl<br/>                     tel. 61 6653031<br/>                     Instytut Informatyki<br/>                     ul. Piotrowo 2, 60-965 Poznań</p>  |  |  |
| <b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>   |  |  |
| 1  | <b>Wiedza:</b>   | <p>Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_W1-2, K_W4, K_W6-15, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału <a href="http://www.fc.put.poznan.pl">www.fc.put.poznan.pl</a></p> <p>W szczególności, 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.</p>                        |
| 2  | <b>Umiejętności:</b>   | <p>Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_U1-2, K_U4, K_U7-8, K_U14-20, K_U22-23, K_U26, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału <a href="http://www.fc.put.poznan.pl">www.fc.put.poznan.pl</a></p> <p>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.</p>          |
| 3  | <b>Kompetencje społeczne</b>   | <p>Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_K1-9, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału <a href="http://www.fc.put.poznan.pl">www.fc.put.poznan.pl</a></p> <p>Powinien również rozumieć konieczność poszerzania swoich kompetencji.</p> <p>Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.</p> |
| <b>Cel przedmiotu:</b>   |  |  |
| <p>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.</p> <p>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.</p> |  |  |
| <b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>  |  |  |
| <b>Wiedza:</b>   |  |  |

|   |
|---|
| <p>1. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: budowa wybranego silnika grafiki, symulacja komputerowa podstawowych zjawisk fizycznych, języki programowania kart graficznych - [K_W5]</p> <p>2. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce związanych przede wszystkim z problemami grafiki trójwymiarowej - [K_W6]</p> <p>3. zna podstawowe techniki animacji modeli 3D, zna przykładowe narzędzie stosowane do modelowania obiektów trójwymiarowych, rozumie problemy związane z wykorzystaniem we własnej aplikacji bibliotek języka C++ oraz języków skryptowych, - [K_W8]</p>  |
| <p><b>Umiejętności:</b></p> <p>1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie, - [K_U1]</p> <p>2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, - [K_U5]</p> <p>3. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne - [K_U9]</p> <p>4. potrafi ? przy formułowaniu i rozwiązywaniu zadań inżynierskich ? integrować wiedzę zarówno z różnych obszarów informatyki takich jak: programowanie obiektowe, metody sztucznej inteligencji, programowanie sieciowe, przetwarzanie danych obiektów trójwymiarowych, jak i innych dyscyplin naukowych takich jak fizyka. - [K_U10]</p> <p>5. potrafi formułować i testować hipotezy związane z problemami inżynierskimi i prostymi problemami badawczymi - [K_U12]</p> <p>6. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13]</p> <p>7. potrafi ? stosując koncepcyjnie nowe metody rozwiązywać złożone zadania dotyczące integracji świata grafiki i fizyki oraz projektować podstawowe skrypty sztucznej inteligencji, potrafi programować podstawowe moduły programowalnego potoku grafiki trójwymiarowej - [K_U25]</p> |
| <p><b>Kompetencje społeczne:</b></p> <p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]</p> <p>2. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K_K6]</p>  |

|  |
|--|
| <p style="text-align: center;"><b>Sposoby sprawdzenia efektów kształcenia</b></p> <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> |
| <p style="text-align: center;"><b>Treści programowe</b></p>  |

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.

Tematyka gier dwuwymiarowych na podstawie biblioteki XNA. Sposoby animowania obiektów, podstawowa pętla gry, zarządzanie zasobami.

Problematyka symulacji fizyki 2D na podstawie silnika Farseer Physics działającego z biblioteką XNA, problem ujednoczenia skali, założenia symulacji fizyki.

Zasady modelowania 3D na przykładzie aplikacji Blender. Zapoznanie się z narzędziem, podstawowe własności modeli, metody tekstuowania, 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).

Właściwości silnika grafiki 3D Ogre3D, podstawowe przekształcenia obiektów, import modeli z programu Blender, animacje, system tworzenia terenu, sposoby realizacji otoczenia (SkyBox, SkyPlane), wyszukiwanie obiektów na scenie.

Integracja silnika grafiki 3D z wybranym silnikiem 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.

Problem atrakcyjności gier, ich grywalności i zasady projektowania gier.

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 trzy główne 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.

Grafika 2D w grach omawiana na przykładzie biblioteki XNA. Studenci realizują zadania związane z zarządzaniem zasobami, animacją duszków (sprite), obsługą urządzeń wejściowych tj. klawiatura, mysz, gamepad. W ramach tego bloku, studenci zapoznają się również z problematyką symulacji fizyki począwszy od prostych pojęć dotyczących grawitacji, systemu kolizji i podstawowych własności fizycznych obiektu takich jak masa, elastyczność itp., do bardziej skomplikowanych zagadnień związanych z oddziaływaniem pomiędzy zadanymi obiektami mechanizmów ich połączeń (joint), siłami tarcia, budową złożonych obiektów oraz reagowaniem na kolizje.

Problemy gier 3D omawiane są na podstawie aplikacji Blender3D oraz w podstawowym stopniu silnika Ogre3D. Na przykładzie Blendera, każdy ze studentów przygotowuje własny model wybranego obiektu 3D i go animuje. Model ten w dalszej kolejności zostanie wykorzystany w silniku Ogre3D. W ramach zajęć pokazane są rozwiązania problemów związanych z konfiguracją środowiska pracy, skali obiektów, ich zależności, budowy sceny, tworzenia terenu, omawiany jest język opisu zasobów. W ramach nauki programowania kart graficznych studenci zapoznają się ze aplikacją FXComposer. Następnie przygotowują funkcje cieniujące zgodnie z modelem oświetlenia Phong'a oraz podstawowe mechanizmy mapowania nierówności.

Metody dydaktyczne:

1. wykład: prezentacje multimedialna, wraz z demonstracją następującego oprogramowania: evalUAtion, Blender, FXComposer
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  | 30            |
| 2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną) | 8<br>5        |
| 3. napisanie programu zaliczeniowego, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)   | 8<br>30       |
| 4. napisanie skryptów sztucznej inteligencji   | 10            |
| 5. udział w wykładach  | 10            |
| 6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron  |               |
| 7. przygotowanie modelu i jego animacji w programie Blender  |               |
| <b>Obciążenie pracą studenta</b>   |               |
| <b>forma aktywności</b>  | <b>godzin</b> |
| <b>ECTS</b>  |               |
| Łączny nakład pracy  | 101           |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem  | 68            |
| Zajęcia o charakterze praktycznym  | 83            |