

| <b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>  |   |   |
|--|---|---|
| Nazwa modułu/przedmiotu<br><b>Programowanie gier</b>   |   | Kod<br><b>1010515321010511655</b>   |
| Kierunek studiów<br><b>Informatyka</b>   | Profil kształcenia<br>(ogólnoakademicki, praktyczny)<br><b>ogólnoakademicki</b> | Rok / Semestr<br><b>1 / 2</b>   |
| Ścieżka obieralności/specjalność<br><b>Zaawansowane technologie internetowe</b>  | Przedmiot oferowany w języku:<br><b>polski</b>                                  | Kurs (obligatoryjny/obieralny)<br><b>obligatoryjny</b>  |
| Stopień studiów:<br><b>II stopień</b>  | Forma studiów (stacjonarna/niestacjonarna)<br><b>niestacjonarna</b>             |   |
| Godziny<br>Wykłady: <b>16</b> Ćwiczenia: - Laboratoria: <b>18</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><br><br><b>nauki techniczne</b>  |   | Podział ECTS (liczba i %)<br><b>4 100%</b><br><br><b>4 100%</b>   |
| <b>Odpowiedzialny za przedmiot / wykładowca:</b><br><br>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   |   |   |
| <b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>   |   |   |
| <b>1</b>   | <b>Wiedza:</b>  | Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, 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><br><br>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.                                       |
| <b>2</b>   | <b>Umiejętności:</b>  | Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, 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><br><br>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.   |
| <b>3</b>   | <b>Kompetencje społeczne</b>  | Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, 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><br><br>Powinien również rozumieć konieczność poszerzania swoich kompetencji.<br><br>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><br>1. Przekazanie studentom podstawowej wiedzy o problemach spotykanych przy programowaniu gier video, w zakresie sposobu działania silników gier na przykładzie środowiska Unreal Engine 4.<br>2. Rozwijanie u studentów umiejętności przygotowania modelu trójwymiarowego danego obiektu wraz z jego animacją.<br>3. Prezentacja podstawowych informacji o mechanizmach tworzenia sztucznej inteligencji w grach komputerowych. |   |   |
| <b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>  |   |   |
| <b>Wiedza:</b>   |   |   |

|  |
|--|
| <p>1. ma zaawansowaną i pogłębioną wiedzę z zakresu elementów składowych gier komputerowych oraz zna środowisko Unreal Engine 4 wykorzystywane do ich implementacji - [K2st_W1]</p> <p>2. ma uporządkowaną i pogłębioną wiedzę z zakresu budowy i komponentów silników gry na przykładzie omawianego oprogramowania Unreal Engine 4. - [K2st_W2]</p> <p>3. ma zaawansowaną wiedzę dotyczącą tworzenia gier video - [K2st_W3]</p> <p>4. 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 - [K2st_W3]</p> <p>5. na przykładzie silnika Unreal Engine 4 ma wiedzę o trendach rozwoju silników gier komputerowych - [K2st_W4]</p> <p>6. zna zaawansowane narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich jakim jest gra video - [K2st_W6]</p>   |
| <b>Umiejętności:</b>   |
| <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 - [K2st_U1]</p> <p>2. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich związanych ze implementacją sztucznej inteligencji na potrzeby gier komputerowych metody symulacyjne - [K2st_U4]</p> <p>3. 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 - [K2st_U5]</p> <p>4. potrafi ocenić przydatność i możliwość wykorzystania nowoczesnych i wciąż rozwijanych narzędzi do tworzenia gier (metod i narzędzi) - [K2st_U6]</p> <p>5. na przykładzie zastosowania silnika Unreal Engine 4 do tworzenia gier 2D i platformy evaluation potrafi dokonać krytycznej analizy rozwiązań technicznych i zaproponować ich ulepszenia - [K2st_U8]</p> <p>6. na przykładzie znajomości modułów i kompleksowych rozwiązań w silniku Unreal Engine 4 potrafi ocenić przydatność innych narzędzi i metod do tworzenia gier komputerowych w tym dostrzec ograniczenia tych metod i narzędzi - [K2st_U9]</p> <p>7. potrafi współdziałać w zespole zarówno na etapie tworzenia projektu końcowego, który może być realizowany w małych zespołach jak i na przykładzie symulacji walki botów sztucznej inteligencji - [K2st_U15]</p> <p>8. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, w tym innych osób - [K2st_U16]</p> |
| <b>Kompetencje społeczne:</b>  |
| <p>1. na przykładzie tempa rozwoju środowiska Unreal Engine 4, rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]</p>  |

### 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 egzamin końcowy realizowany w formie mieszanej: testu jednokrotnego wyboru oraz pytań otwartych z zagadnień przedstawianych w trakcie zajęć. Pytania są punktowane w zależności od ich stopnia trudności. Do uzyskania zaliczenia należy uzyskać co najmniej 50% możliwych do uzyskania punktów. W trakcie egzaminu nie można korzystać z żadnych materiałów.

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- 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,
- ocenę i obronę przez studenta lub małej grupy studentów 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

|  |                            |
|--|----------------------------|
| <p>Program wykładu obejmuje następujące zagadnienia:</p> <p>Problematyka sztucznej inteligencji w grach realizowana jest w oparciu o projekt evalUAtion. 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 UnrealEngine 4. ? omówienie podstawowych okien środowiska, tworzenie projektu i sceny, podstawowych klas obiektów silnika, programowanie za pomocą Blueprints, sztuczna inteligencja.</p> <p>Zasady modelowania 3D na przykładzie aplikacji Blender. Zapoznanie się z narzędziem, podstawowe własności modeli, metody teksturowania, rodzaje oświetlenia, mapowanie nierówności, eksport modelu do silnika Unreal Engine 4. Rodzaje animacji w grafice 3D na przykładzie programu Blender. Pojęcia klatek kluczowych, rodzaje interpolacji, animacja szkieletowa oraz deformacje siatki obiektów (ang. Shape Keys).</p> <p>Omówienie zagadnień dotyczących tworzenia materiałów oraz systemów cząstek w środowisku Unreal Engine 4.</p> <p>Zajęcia laboratoryjne dotyczą następujących zagadnień.</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. W późniejszym etapie, zostają zaprezentowane elementy sztucznej inteligencji w środowisku Unreal Engine 4.</p> <p>Problematyka dotyczące gier 3D przedstawiana jest w środowisku Unreal Egnine 4. Po zapoznaniu się z podstawowymi modułami silnika, studenci poznają podstawowe klasy obiektów w grze, uczą się tworzyć proste modele statyczne oraz animowane postaci w programie Blender. Wykorzystanie przygotowanego modelu do utworzenia postaci w grze. Interakcja postaci z elementami sceny, symulacja fizyki. Jako platformę programistyczną wykorzystuje się skrypty silnika Unreal tj. Blueprints.</p> <p>Zagadnienia związane z interakcją użytkownika z grą tj.: tworzenie interfejsu użytkownika (ekran początkowy, ekran końcowy, inne elementy interfejsu np. punkty, pasek życia itp.) oraz definiowanie sposobu kontroli użytkownika nad grą.</p> <p>Elementy tworzenia realistycznych materiałów oraz prezentacja systemu cząstek (particle system) do tworzenia specjalistycznych efektów takich jak: mgła, dym, ogień itp.</p> <p>Wykorzystanie mechanizmów sztucznej inteligencji dostępnych w silniku UnrealEngine: Navigation Mesh i Behaviour Tree.</p> <p>Metody dydaktyczne:</p> <ol style="list-style-type: none"> <li>1. wykład: prezentacje multimedialna, wraz z demonstracją następującego oprogramowania: evalUAtion, Blender, Unreal Engine 4</li> <li>2. ćwiczenia laboratoryjne: rozwiązywanie zadań obejmujących wykorzystanie wybranych technologii np. przygotowanie fragmentu gry, rozbudowa otrzymanego projektu o wybrane elementy itd., 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 &amp; Associates, 2003</li> <li>3. Programming Game AI by Example, Buckland, M., Jones &amp; Bartlett Publishers, 2004</li> <li>4. Język Cg. Programowanie grafiki w czasie rzeczywistym, Fernando, R., Kilgard, M.J., Helion, 2003</li> <li>5. Blender. Kompendium, Kuklo, K., Kolmaga, J., Helion, 2007</li> <li>6. Blender 2.69 : architektura i projektowanie , Chlipalski, Piotr, Helion, 2014.</li> <li>7. 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> <li>3. Mathematics for 3D Game Programming and Computer Graphics, Third Edition, Lengyel, E., Course Technology PTR, 2011</li> <li>4. Shadery : zaawansowane programowanie w GLSL, Sobiesiak, Karol., Sydow, Piotr. PWN, 2015</li> </ol>  |                            |
| <p><b>Bilans nakładu pracy przeciętnego studenta</b></p>   |                            |
| <p><b>Czynność</b></p>   | <p><b>Czas (godz.)</b></p> |

|  |               |
|--|---------------|
| 1. udział w zajęciach laboratoryjnych / ćwiczeniach  | 16            |
| 2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną) | 4<br>25       |
| 3. napisanie programu zaliczeniowego, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)   | 5<br>16       |
| 4. napisanie skryptów sztucznej inteligencji   | 15            |
| 5. udział w wykładach  | 10            |
| 6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 150 stron  | 10            |
| 7. przygotowanie modelu i jego animacji w programie Blender  |               |
| 8. przygotowanie do egzaminu i obecność na egzaminie: 8 godz. + 2 godz.  |               |
| <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  | 38            |
| Zajęcia o charakterze praktycznym  | 56            |