



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

### Bogate aplikacje internetowe

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Gry i technologie internetowe

Poziom studiów

Studia II stopnia

Forma studiów

Studia stacjonarne

Rok/semestr

1/1

Profil studiów

Ogólniakademicki

Język oferowanego przedmiotu

Język polski

Wymagalność

Przedmiot obligatoryjny

### Liczba godzin

Wykład

15

Ćwiczenia

Laboratoria

15

Projekty/seminaria

Inne (np. online)

### Liczba punktów ECTS 2

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Andrzej Urbański

Instytut Informatyki PP

ul. Piotrowo 2, 60-965 Poznań

e-mail: [andrzej.urbanski@put.poznan.pl](mailto:andrzej.urbanski@put.poznan.pl)

### Wymagania wstępne

Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K\_W1-2, K\_W4, K\_W6-15, K\_U1-2, K\_U4, K\_U7-8, K\_U14-20, K\_U22-23, K\_U26, K\_K1-9, weryfikowane w procesie rekrutacji na studia 2 stopnia – efekty te prezentowane są w serwisie internetowym wydziału [www.fc.put.poznan.pl](http://www.fc.put.poznan.pl)

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z programowania OpenGL. Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu podstaw programowania gier komputerowych i grafiki oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również mieć gotowość do podjęcia współpracy w ramach zespołu.

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 z programowania grafiki 3D, w zakresie programowania WebGL dla różnych potrzeb
2. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z programowaniem 3D



3. Kształtowanie u studentów umiejętności pracy zespołowej w zakresie programowania 3D

### Przedmiotowe efekty uczenia się

#### Wiedza

1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie technologii sieciowych, języków i paradygmatów programowania grafiki
2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: programowanie grafiki, programowanie gier, programowanie multimedialnych stron internetowych
3. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z zakresu programowania grafiki i gier

#### Umiejętności

1. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia w zakresie programowania w środowisku WebGL
2. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych
3. potrafi zaproponować ulepszenia (usprawnienia) istniejących rozwiązań technicznych

#### Kompetencje społeczne

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe
2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia
3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania

### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

#### 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 i umiejętności wykazanych na kolokwium pisemnym o charakterze problemowym (student może korzystać z dowolnych materiałów dydaktycznych) 5 pytań, 10pkt, minimum 5pkt na ocenę dostateczną
  - omówienie wyników kolokwium,
- b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:



- ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole,
- ocenę i „obronę” przez studenta sprawozdania z realizacji projektu,

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- uwagi związane z udoskonaleniem materiałów dydaktycznych,
- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

### Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Przypomnienie wiadomości z programowania OpenGL. Schemat programu gry komputerowej, a szkielet jej kodu w WebGL. Programowanie w czystym WebGL z ręcznym kodowaniem grafiki. Przygotowywanie grafiki w programach i kod w WebGL do jej importu. Użycie edytora graficznego Coppercube. Dokładanie kodu zdarzeń w JavaScript. Środowisko Three.js i jego użycie w programowaniu gier komputerowych. Programowanie kamery komputera dla sterowania grą za pośrednictwem gestów.

Zajęcia laboratoryjne prowadzone są w formie piętnastu godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych 1-godzinną sesją instruktazową na początku semestru. Ćwiczenia realizowane są przez 2-osobowe zespoły studentów.

Program laboratorium obejmuje następujące zagadnienia:

Programowanie w WebGL. Tworzenie grafiki w środowisku WebGL dla zadanych prostych przykładów. Wzbogacanie mechaniki gry o kod w JavaScript.

### Metody dydaktyczne

1. Wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy.

2. Ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, dyskusja, praca w zespole.

### Literatura

#### Podstawowa

Jacob Seidelin [HTML5. Tworzenie gier](#), Helion, Gliwice, 2012

Karl Bunyan [HTML5 : tworzenie gier z wykorzystaniem CSS i JavaScript](#), Helion, 2016.

#### Uzupełniająca

Tony Parisi [Aplikacje 3D : przewodnik po HTML5, WebGL i CSS3](#), Helion, 2015.

Sobiesiak, Karol., Sydow, Piotr. [Shadery : zaawansowane programowanie w GLSL](#) PWN, 2015.

### Bilans nakładu pracy przeciętnego studenta

|   | Godzin | ECTS     |
|---|--------|----------|
| Łączny nakład pracy   | 75     | 2        |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem   | 38     | 1        |
| Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do kolokwium, wykonanie projektu) | 37     | <b>1</b> |