

OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Programowanie gier komputerowych / Computer Games Programming
2.	Dyscyplina Informatyka techniczna i telekomunikacja – 2, Informatyka - 2
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Wydział Fizyki i Astronomii
5.	Kod przedmiotu/modułu 24-ISSP-S1-E7-PGK
6.	Rodzaj przedmiotu/modułu (<i>obowiązkowy lub do wyboru</i>) do wyboru
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka stosowana i systemy pomiarowe
8.	Poziom studiów (<i>I lub II stopień, jednolite studia magisterskie, studia doktoranckie</i>) I stopień
9.	Rok studiów (<i>jeśli obowiązuje</i>) 4
10.	Semestr (<i>zimowy lub letni</i>) zimowy
11.	Forma zajęć i liczba godzin Laboratorium komputerowe 30 godz.
12.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia Michał Nowak, magister inżynier
13.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Podstawy programowania Programowanie C++ Podstawy grafiki / animacji komputerowej
14.	Cele przedmiotu Przedmiot zaznajamia studentów z podstawowymi zasadami programowania gier komputerowych.
15.	Treści programowe Studenci poznają środowisko developerów gier komputerowych. Prowadzą prosty projekt wg wskazówek prowadzącego stosując zasady i konstrukcje używane w praktycznym programowaniu gier (np. układ klas, logika gry, odpowiedni poziom

	abstrakcji kodu itp.). Wynikiem kursu jest gra komputerowa (lub kilka prostych mechanik gry) stworzona własnoręcznie w trakcie zajęć.	
16.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <ul style="list-style-type: none"> • Zna zasady programowania strukturalnego oraz obiektowego, wybrane języki programowania (skryptowe, kompilowane, graficzne), a także odpowiednie środowiska programistyczne i narzędzia do tworzenia, kontroli wersji, testowania i dystrybucji gier. • Zna podstawowe zasady dotyczące działania gier komputerowych. Zna sposób konstruowania klas, ich zależności i budowy podstawowego silnika do gier. • Rozumie istotę i znaczenie tzw. mechaniki gry, rozumie istotę designu w tworzeniu tego typu oprogramowania. • Posiada wiedzę o sposobach dystrybucji gotowych gier. • Zna źródła materiałów (darmowych i komercyjnych), których można użyć w grach komputerowych. <p>Umiejętności</p> <ul style="list-style-type: none"> • Potrafi uczyć się samodzielnie; sprawnie wyszukuje i wykorzystuje informacje niezbędne do poznania nowego zagadnienia lub rozwiązania problemu. • Umie wprowadzać elementy podnoszące tzw. grywalność do swojego projektu. • Umie zaprogramować prostą grę opartą o własny, prosty silnik (czyli układ klas C++ służących do wyświetlania modeli, poruszania, prostej kolizji). • Potrafi wykorzystać źródła materiałów do gier znajdujący się w internecie. <p>Kompetencje społeczne</p> <ul style="list-style-type: none"> • Jest otwarty na współpracę i wymianę myśli; potrafi pracować w grupie; jest gotów podejmować rzeczową dyskusję i aktywnie uczestniczyć w debacie. • Potrafi myśleć kreatywnie i działać twórczo; jest otwarty na nowe pomysły i nowe technologie. • Jest świadomy potrzeby samorozwoju oraz podnoszenia kompetencji zawodowych i osobistych. 	<p>Symbole odpowiednich kierunkowych efektów uczenia się:</p> <p>I1_W04</p> <p>I1_U17</p> <p>I1_K01</p> <p>I1_K05</p>
17.	<p>Literatura obowiązkowa i zalecana (<i>źródła, opracowania, podręczniki itp.</i>)</p> <p>Literatura zalecana:</p> <ul style="list-style-type: none"> • Perełki programowania gier. Vademecum profesjonalisty. Tom 1 – 6 • Game Engine Architecture, Gregory Jason • Game Programming Patterns, Robert Nystrom • Skuteczny nowoczesny C++. 42 sposoby lepszego posługiwania się 	

	<p>językami C++11 I C++14, Scott Meyers</p> <ul style="list-style-type: none"> C++. 50 efektywnych sposobów na udoskonalenie Twoich programów, Scott Meyers 	
18.	<p>Metody weryfikacji zakładanych efektów uczenia się:</p> <p>Ocena projektów wykonanych w ramach laboratorium</p>	
19.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu:</p> <p>Ocena projektów wykonanych w ramach laboratorium</p>	
20.	<p>Nakład pracy studenta/doktoranta</p>	
	forma działań studenta/doktoranta	liczba godzin na realizację działań
	Zajęcia (wg planu studiów) z prowadzącym:	
	- wykład:	- 0
	- konwersatorium:	- 0
	- laboratorium:	- 30
- inne:	- 0	
Praca własna studenta, doktoranta (w tym udział w pracach grupowych):		
- przygotowanie do zajęć:	- 20	
- opracowanie wyników:	- 5	
- czytanie wskazanej literatury:	- 15	
- przygotowanie prac/wystąpień/projektów:	- 30	
- napisanie raportu z zajęć:	- 0	
- przygotowanie do sprawdzianów i egzaminu:	- 0	
Łączna liczba godzin	100	
Liczba punktów ECTS	4	