



## Projektowanie aplikacji VR Karta opisu przedmiotu

### Informacje podstawowe

<b>Kierunek studiów</b> Informatyka Społeczna	<b>Cykl dydaktyczny</b> 2026/2027	
<b>Specjalność</b> Wszystkie	<b>Kod przedmiotu</b> HIFSS.II4.15576.26	
<b>Jednostka organizacyjna</b> Wydział Humanistyczny	<b>Języki wykładowe</b> polski	
<b>Poziom kształcenia</b> Studia magisterskie II stopnia	<b>Obligatoryjność</b> Obowiązkowy	
<b>Forma studiów</b> Stacjonarne	<b>Blok zajęciowy</b> Przedmioty kierunkowe	
<b>Profil studiów</b> Praktyczny	<b>Przedmiot powiązany z badaniami naukowymi</b> Tak	
<b>Koordinator przedmiotu</b>	Jowita Guja	
<b>Prowadzący zajęcia</b>	Jowita Guja	
<b>Okres</b> Semestr 3	<b>Forma zaliczenia</b> Zaliczenie	<b>Liczba punktów ECTS</b> 3
	<b>Forma prowadzenia i godziny zajęć</b> Ćwiczenia laboratoryjne: 30	

### Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
<b>Wiedzy - Student zna i rozumie:</b>			
W1	Student zna metody i narzędzia programowania funkcjonalności środowisk wirtualnych w języku C.	IFS2P_W02	Aktywność na zajęciach, Wykonanie ćwiczeń, Projekt

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
W2	Ma wiedzę z zakresu projektowania interaktywnych środowisk wirtualnych.	IFS2P_W01	Aktywność na zajęciach, Wykonanie ćwiczeń, Wykonanie projektu
<b>Umiejętności - Student potrafi:</b>			
U1	Potrafi zaprojektować i zbudować aplikacje rzeczywistości poszerzonej i wirtualnej.	IFS2P_U04	Aktywność na zajęciach, Wykonanie ćwiczeń, Wykonanie projektu
<b>Kompetencji społecznych - Student jest gotów do:</b>			
K1	Potrafi myśleć innowacyjnie i kreatywnie przy projektowaniu i wdrażaniu aplikacji rzeczywistości rozszerzonej i/lub wirtualnej.	IFS2P_K01	Aktywność na zajęciach

### Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Student zna i potrafi wykonać programowanie funkcjonalności środowisk wirtualnych w języku C# na silniku Unity oraz umie zaprojektować aplikacje rzeczywistości poszerzonej i wirtualnej.

### Nakład pracy studenta

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Ćwiczenia laboratoryjne	30
Przygotowanie do zajęć	13
Samodzielne studiowanie tematyki zajęć	15
Dodatkowe godziny kontaktowe	2
Przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	15
<b>Łączny nakład pracy studenta</b>	<b>Liczba godzin</b> 75
<b>Liczba godzin kontaktowych</b>	<b>Liczba godzin</b> 30

\* godzina (lekcyjna) oznacza 45 minut

### Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
1.	Programowanie środowisk wirtualnych w języku C# na silniku Unity: fizyka interaktywnych środowisk wirtualnych, modularność, skalowalność, sprawdzone praktyki doskonalenia umiejętności programistycznych zmierzającego do pisania czystego kodu dla interaktywnych środowisk wirtualnych, programistyczne techniki optymalizacji, zagadnienia sieciowe w programowaniu środowisk wirtualnych, programowanie sztucznej inteligencji, testowanie interaktywnych środowisk wirtualnych, testy jednostkowe, zabezpieczenia interaktywnych środowisk wirtualnych	W1, W2, U1, K1	Ćwiczenia laboratoryjne

## Informacje rozszerzone

### Metody i techniki kształcenia :

Rodzaj zajęć	Metody zaliczenia	Warunki zaliczenia przedmiotu
Ćwiczenia laboratoryjne	Aktywność na zajęciach, Wykonanie ćwiczeń, Wykonanie projektu, Projekt	

### Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

Podstawą zaliczenia jest projekt. Zasady i forma zaliczenia w drugim (w sesji) i trzecim (w sesji poprawkowej) terminie pozostaje bez zmian. Obecności są wymagane w ramach nieobecności student/ka musi w ramach dyżuru zaliczyć wymagane ćwiczenia i/lub literaturę.

### Sposób obliczania oceny końcowej

100% Projekt realizowany w formie ćwiczeń na zajęciach, który będzie polegał na zbudowaniu przez studentów aplikacji rzeczywistości poszerzonej lub/i wirtualnej.

### Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

Zaległości student może nadrobić w oparciu o literaturę zaleconą przez wykładowcę. Powstałe zaległości student zalicza w terminie ustalonym z wykładowcą.

## Wymagania wstępne i dodatkowe

Znajomość języka programowania C#

### Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Ćwiczenia laboratoryjne: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych.

## Literatura

### Obowiązkowa

1. zalecana literatura oraz oprogramowanie zostanie studentom dostarczone na zajęciach.

## Badania i publikacje

### Publikacje

1. Karol Matyasik
2. Wielki pasjonat gier wszystkich rodzajów, od karcianych, po komputerowe. Od zawsze interesował się mechanikami rozgrywki i produkcją gier.
3. Posiada wszechstronne doświadczenie w tworzeniu gier od podstaw w popularnych silnikach - Unity i Unreal Engine 4, zdobyte podczas pracy zawodowej i własnych projektów.
4. Tytuł magistra inżyniera telekomunikacji otrzymał w 2017 roku na Akademii Górniczo Hutniczej, za pracę "Emotion recognition from speech signals for video games". 2014 - 2015 pracował w firmie AON jako Information security analyst (Ochrona informacji, networks, firewalle itp)
5. W 2016 r. jako uczestnik programu Vulcanus, pracował w Advanced Technology Devision w firmie Square-Enix w Tokyo przy technologiach służących do produkcji gier.
6. Od 2017 roku pracuje jako gameplay designer w Krakowskim oddziale CD PROJEKT RED nad Cyberpunkiem 2077.

## Kierunkowe efekty uczenia się

Kod	Treść
IFS2P_K01	jest przedsiębiorczy, potrafi myśleć innowacyjnie i kreatywnie, łączyć schematy myślowe przy projektowaniu i wdrażaniu rozwiązań oraz ma zdolność myślenia strategicznego
IFS2P_U04	potrafi formułować i testować hipotezy związane z prostymi problemami wdrożeniowymi; umie zaprojektować zaawansowane rozwiązania techniczne (w formie urządzeń, obiektów, systemów lub procesów) oraz ma umiejętności pozwalające na ich realizację
IFS2P_W01	ma wiedzę z zakresu automatyki i robotyki, projektowania komunikacji człowiek-komputer oraz projektowania interaktywnych środowisk wirtualnych
IFS2P_W02	ma zaawansowaną wiedzę z zakresu metod i technik programowania, konstrukcji i ekstrakcji baz danych oraz grafiki komputerowej oraz projektowania interfejsów, interakcji i doświadczeń użytkownika