



Introduction to Game Engines

Course description sheet

Basic information

Field of study Designing Virtual Spaces and Games	Didactic cycle 2025/2026
Major -	Course code WIPPWGS.I2.17391.25
Organisational unit Faculty of Computer Science	Lecture languages Polish
Study level First-cycle studies	Mandatoriness Obligatory
Form of study Full-time studies	Block Core Modules
Profile General academic	Course related to scientific research No
Course coordinator	Bernard Maj
Lecturer	Bernard Maj
Period Semester 2	Method of verification of the learning outcomes Completing the classes
	Activities and hours Lectures: 14 Laboratory classes: 28
	Number of ECTS credits 4

Goals

C1	Zapoznanie studentów z podstawowymi koncepcjami i architekturą popularnych silników gier komputerowych, takich jak Unity, Unreal Engine, CryEngine i Godot. Przekazanie wiedzy z zakresu tworzenia prototypów gier 2D i 3D, obejmującej programowanie skryptowe, zarządzanie assetami, fizykę oraz interfejs użytkownika. Uświadomienie słuchaczom problemów związanych z optymalizacją wydajności oraz projektowaniem mechanik gry w środowisku edytorów wizualnych.
----	---

Course's learning outcomes

Code	Outcomes in terms of	Learning outcomes prescribed to a field of study	Methods of verification
Knowledge - Student knows and understands:			
W1	Student zna i rozumie pojęcie silnika gier	PPWG1A_W04	Activity during classes, Execution of laboratory classes
W2	Student zna dobre i słabe strony różnych silników gier. Wie, który z nich zastosować do realizacji gry o zadany scenariusz.	PPWG1A_W04	Activity during classes, Execution of laboratory classes
Skills - Student can:			
U1	Student potrafi wykorzystać różne silniki gier do tworzenia prostych aplikacji.	PPWG1A_U07	Activity during classes, Execution of laboratory classes
Social competences - Student is ready to:			
K1	Student potrafi współpracować w zespole tworzącym oprogramowanie	PPWG1A_K02	Activity during classes, Execution of laboratory classes

Program content ensuring the achievement of the learning outcomes prescribed to the module

Wprowadzenie: różnorodność silników gier. Przetwarzanie projektów w różnych silnikach gier. Tworzenie prostych aplikacji.

Student workload

Activity form	Average amount of hours* needed to complete each activity form
Lectures	14
Laboratory classes	28
Preparation for classes	25
Realization of independently performed tasks	25
Preparation of project, presentation, essay, report	15
Student workload	Hours 107
Workload involving teacher	Hours 42

* hour means 45 minutes

Program content

No.	Program content	Course's learning outcomes	Activities
1.	<p>#Wprowadzenie do silników gier Czym jest silnik gier: runtime, edytor, narzędzia i typowe podsystemy (grafika, fizyka, audio, UI). Przegląd Unity, Unreal Engine, CryEngine i Godot – ogólny profil i typowe zastosowania.</p> <p>#Podstawy pracy w edytorze Zakładanie projektu, podstawowe ustawienia, nawigacja po edytorze (widoki, sceny/poziomy). Obiekty w świecie: transformacje (pozycja/obrót/skala), hierarchia, grupowanie. Play mode/testowanie.</p> <p>#Tworzenie Sceny i obiektów gry Co to jest scena/level i jak ją organizować (podział na elementy statyczne i dynamiczne). Podejście komponentowe: obiekt + dołączone zachowania (idea wspólna dla większości silników). Prefab/podobne mechanizmy jako sposób ponownego użycia obiektów.</p> <p>#Skrypty i logika Jak powstaje logika gry: skrypt tekstowy lub narzędzia wizualne (na poziomie koncepcji). Zdarzenia i aktualizacja w czasie: input, prosty ruch, prosta interakcja. Debugowanie podstawowe: logi, inspektor/parametry.</p> <p>#Assets i materiały Co to są assets (modele, tekstury, dźwięki) i jak je importować oraz porządkować w projekcie. Podstawy materiałów i tekstur: przypisanie do obiektu, podstawowe parametry „wizualne”. Dobre praktyki: foldery, nazewnictwo i przygotowanie pod pracę zespołową.</p> <p>#Podstawowe elementy „grywalności” Kolizje i prosta fizyka: obiekty statyczne/dynamiczne, triggery, raycast. UI w wersji podstawowej: HUD/licznik, prosty ekran pauzy lub menu. Audio w wersji podstawowej: odtwarzanie SFX i podstawy audio 3D.</p> <p>#Budowanie i porównanie silników Minimalny „build” na PC: konfiguracje, uruchomienie wersji testowej, checklisty. Proste profilowanie/diagnoza problemów (co mierzyć i jak wyciągać wnioski na poziomie podstawowym). Podsumowanie porównawcze: szybkość prototypu, łatwość nauki, workflow edytora, problemy z silnikami..</p>	W1, W2	Lectures

No.	Program content	Course's learning outcomes	Activities
2.	<p>Instalacja i konfiguracja środowiska (Unity, Unreal Engine, CryEngine, Godot + IDE). Repozytorium Git, struktura projektu, konwencje nazewnictwa, wspólny pakiet assetów do porównań. Stworzenie prostych gier 2D i 3D w każdym z silników wykorzystując poniższe działania Ustalenie kryteriów porównania silników (czas wykonania, liczba kroków w edytorze, ergonomia, typowe problemy). utworzenie projektu i sceny, nawigacja po edytorze, podstawowe obiekty i transformacje. kamera, input i prosty ruch postaci/obiektu. kolizje i triggery, proste interakcje (np. podnoszenie obiektu, przełącznik). prosty system punktów/stanu gry, podstawowe debugowanie (logi/inspektor). podstawowe UI (HUD: licznik/zdrowie) oraz audio (SFX). przygotowanie i wykonanie builda na PC, podstawowe ustawienia build/paczki.</p> <p>Tworzenie własnej prostej gry w wybranym przez siebie silniku.</p>	U1, K1	Laboratory classes

Extended information/Additional elements

Teaching methods and techniques :

Lecture, Lectures, Workshop, E-learning, Group work, Discussion

Activities	Methods of verification	Credit conditions
Lectures	Activity during classes, Execution of laboratory classes	
Lab. classes	Activity during classes, Execution of laboratory classes	

Additional info

Wykłady mogą odbywać się w sposób stacjonarny (w salach) lub w trybie zdalnym na platformie MS Teams.

Conditions and the manner of completing each form of classes, including the rules of making retakes, as well as the conditions for admission to the exam

Ocena końcowa jest równa ocenie z zaliczenia zajęć laboratoryjnych, a ta wyznaczana jest skali określonej w regulaminie AGH na podstawie sumy punktów uzyskiwanych za uczestnictwo w zajęciach i prezentowane rozwiązania postawionych problemów.

Method of determining the final grade

Aby uzyskać pozytywną ocenę końcową niezbędne jest uzyskanie pozytywnej oceny z zaliczenia zajęć laboratoryjnych – ocena końcowa jest taka jak ocena z zaliczenia zajęć laboratoryjnych.

Manner and mode of making up for the backlog caused by a student justified absence from classes

Zaległości powstałe wskutek nieobecności studenta na zajęciach warsztatowych można uzupełniać poprzez realizację ćwiczeń i dodatkowych zadań po wcześniejszym uzgodnieniu z prowadzącym.

Prerequisites and additional requirements

brak

Rules of participation in given classes, indicating whether student presence at the lecture is obligatory

Wykład: obecność obowiązkowa, studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Laboratoria: obecność obowiązkowa, studenci wykonują ćwiczenia warsztatowe zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane podczas zajęć. Zaliczenie zajęć odbywa się na podstawie wyniku prac w postaci rozwiązania postawionego problemu.

Literature

Obligatory

1. Jacek Ross, Unity i C#. Praktyka programowania gier, Helion 2020

Optional

1. Kumsal Obuz, Game Development with Blender and Godot. Leverage the combined power of Blender and Godot for building a point-and-click adventure game, Helion 2022
2. Aram Cookson, Ryan Dowling Soka, Clinton Crumpler, Unreal Engine w 24 godziny. Nauka tworzenia gier, Helion 2017
3. Sam Howels, Richard G Marcoux, Riham Mohamed F Aly Aly Toulan, Samuel Howels, Chris Goodswen, Riham Toulan, Richard Marcoux III, CRYENGINE Game Development Blueprints. Perfect the art of creating CRYENGINE games through exciting, hands-on game development projects, Helion 2015

Learning outcomes prescribed to a field of study

Code	Content
PPWG1A_K02	Jest przygotowany do współdziałania i pracy z innymi osobami w ramach zespołu projektowego gier, umie zorganizować pracę własną i zespołową w ramach realizacji wspólnych zadań i projektów.
PPWG1A_U07	Potrafi stosować narzędzia z zakresu projektowania graficznego, grafiki komputerowej i komunikacji wizualnej.
PPWG1A_W04	Zna i rozumie trendy rozwojowe różnych gatunków gier, modeli wirtualnej rzeczywistości i technik grafiki komputerowej i animacji.