



UNIWERSYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Implementacja gry komputerowej Sylabus zajęć

Informacje podstawowe

Kierunek studiów Informatyka Specjalność - Jednostka organizacyjna Wydział Matematyki i Informatyki Poziom studiów studia inżynierskie pierwszego stopnia Forma studiów studia stacjonarne Profil studiów profil ogólnoakademicki		Cykl dydaktyczny 2023/24 Kod zajęć 06INFS.320S.03134.23 Języki wykładowe polski Obligatoryjność Fakultatywny Blok zajęciowy Przedmioty specjalnościowe
Koordynator zajęć	Wojciech Kowalewski	
Prowadzący zajęcia	Wojciech Kowalewski, Andrzej Kokosza	
Okres Semestr 6	Forma zajęć / liczba godzin / forma zaliczenia • Laboratorium: 60, Zaliczenie z oceną	Liczba punktów ECTS 5

Cele kształcenia dla zajęć

Kod	Cel
C1	Celem kursu jest prezentacja i konstrukcja podstawowych modułów składających się na typową grę RPG. Ten scenariusz zostanie zrealizowany za pomocą silnika Unreal Engine 5.

Wymagania wstępne

Wiedza z kursów: Grafika komputerowa, Metody renderingu i Modelowanie geometryczne. Umiejętność programowania w C++ na poziomie obiektowym.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	Zna interfejs graficzny silnika UE5	INF_K3_W03, INF_K3_W10	Projekt
W2	Zna możliwości i sposób wykorzystania języka skryptowego Blueprint	INF_K3_W03, INF_K3_W10	Projekt
W3	Zna sposób tworzenia projektu gry w języku Blueprint	INF_K3_W02, INF_K3_W10	Projekt
W4	Zna sposób tworzenia projektu gry w języku C++	INF_K3_W03, INF_K3_W04_inz, INF_K3_W05, INF_K3_W10	Projekt
W5	Zna sposoby importowania assetów do sceny gry	INF_K3_W03, INF_K3_W10	Projekt
W6	Zna podstawowe klasy specyfikacji logiki aktora oraz jego animacji	INF_K3_W03, INF_K3_W05, INF_K3_W10	Projekt
W7	Zna koncepcję brył otaczających jako mechanizmu wykrywania kolizji pomiędzy obiektami	INF_K3_W01, INF_K3_W05, INF_K3_W10	Projekt
W8	Zna podstawowe sposoby wykrywania kolizji i reagowania na nie - umie wyprowadzić ich postać z różnych praw fizyki	INF_K3_W10	Projekt
Umiejętności - Student/ka:			
U1	Potrafi debugować projekt C++ w środowisku Visual Studio	INF_K3_U05_inz	Projekt
U2	Potrafi zaimportować postać z pliku FBX	INF_K3_U05_inz	Projekt
U3	Potrafi zdefiniować logikę aktora wykorzystując adekwatne do tego klasy	INF_K3_U05_inz	Projekt
U4	Potrafi animować aktora na scenie	INF_K3_U04_inz, INF_K3_U05_inz	Projekt
U5	Potrafi użyć mechanizm dziedziczenia klas w językach Blueprint oraz C++	INF_K3_U04_inz, INF_K3_U05_inz	Projekt
U6	Potrafi sterować aktorami na scenie za pomocą typowych urządzeń wejścia	INF_K3_U05_inz	Projekt
U7	Potrafi wykrywać kolizje pomiędzy obiektami, a także wykrywać przeszkody na ścieżce ruchu	INF_K3_U05_inz	Projekt
U8	Potrafi zdefiniować mechanizm kontrolowania poziomu zdrowia postaci w grze	INF_K3_U04_inz, INF_K3_U05_inz	Projekt
U9	Potrafi zbudować dowolnie złożony interfejs graficzny dla gry z wykorzystaniem dostępnych w UE5 narzędzi	INF_K3_U05_inz, INF_K3_U06_inz	Projekt
U10	Potrafi dodać do gry efekty dźwiękowe	INF_K3_U04_inz, INF_K3_U05_inz	Projekt

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
U11	Potrafi wykorzystać system animacji cząsteczkowej w UE5 do generowania efektów specjalnych	INF_K3_U01_inz, INF_K3_U05_inz	Projekt
U12	Potrafi precyzyjnie animować postać gry z uwzględnieniem szkieletu oraz cech fizycznych	INF_K3_U04_inz, INF_K3_U05_inz, INF_K3_U07_inz	Projekt
U13	Potrafi dodać do aktorów sceny logikę sztucznej inteligencji	INF_K3_U01_inz, INF_K3_U04_inz	Projekt

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Wprowadzenie do Unreal Engine 1. Wymagania techniczne 2. Tworzenie projektu 3. Omówienie interfejsu 4. Okno edytora 5. Okno wizualizacji 6. Dodawanie i usuwanie aktorów sceny, przekształcenia aktorów 7. Różnice pomiędzy klasami Blueprint i C++ - dodawanie do sceny aktora Blueprint 8. Struktura okien edytora Blueprint 9. Pierwsze spotkanie z Blueprint Visual Scripting - okno edycji węzłów (Event Graph) - typy węzłów i sposoby ich łączenia 10. BeginPlay i Tick jako przykłady elementarnych węzłów zdarzeniowych 11. Klasa ThirdPersonCharacter jako przykład klasy reprezentującej postać w grze - podstawowe komponenty postaci - mesh i materiał 12. Podstawy tworzenia i edycji materiałów	W1, W2, W3	Laboratorium
2.	Praca z Unreal Engine 1. Tworzenie pustego projektu C++ 2. Katalog Content i jego struktura 3. Debugowanie projektu w Visual Studio 2022 4. Import postaci z pliku FBX, jako przykład importowania assetów 5. Wstęp do logiki gry - klasa Game Mode i jej składowe. Podstawowe zdarzenia klasy Game Mode i jej rola w grze sieciowej oraz w konstrukcji poziomów gry. 6. Klasa Pawn jako podstawowa klasa specyfikacji logiki aktora 7. Klasa Player Controller jako podstawowa klasa interakcji z aktorem 8. Praca z animacjami: ◦ Animacje w Blueprint: Event Graph i Anim Graph ◦ Rola State Machine w organizacji potoku animacji	W4, W5, W6, U1, U2, U3, U4	Laboratorium
3.	Klasa Character i jej składowe. Wykorzystanie mechanizmu dziedziczenia w językach Blueprint oraz C++.	U5	Laboratorium

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
4.	Legacy Input System i Enhanced Input System jako dwa mechanizmy obsługi zdarzeń 'wejścia'. Sterowanie ruchem aktorów za pomocą klawiatury i myszy.	U6	Laboratorium
5.	Wykrywanie i obsługa kolizji. Wykrywanie przeszkód na ścieżkach ruchu (Line Traces). Definiowanie brył otaczających dla aktorów.	W7, W8, U7	Laboratorium
6.	Definiowanie komponentu zdrowia postaci z wykorzystaniem narzędzi Blueprint oraz cech języka C++	U8	Laboratorium
7.	Tworzenie interfejsu graficznego dla gry (User Interface i Heads Up Display) z wykorzystaniem narzędzia Unreal Motion Graphics (UMG)	U9	Laboratorium
8.	Dodawanie do gry efektów dźwiękowych (klasy: Sound Base, Sound Wave, Sound Cue, Sound, Sound Attenuation, Audio Component)	U10	Laboratorium
9.	Wstęp do generowania efektów specjalnych w grze (VFX) - zastosowanie systemu animacji cząstek	U11	Laboratorium
10.	Szczegółowa animacja postaci z wykorzystaniem komponentu Character Movement. Deformacja siatki szkieletowej. Wykorzystanie assetów Blend Spaces.	U12	Laboratorium
11.	Rozszerzenie kontroli aktorów sceny na logikę sztucznej inteligencji - wykorzystanie AI Controlers, Blackboards i Behaviors Trees.	U13	Laboratorium
12.	Tworzenie własnego projektu gry	W1, W2, W3, W4, W5, W6, W7, W8, U1, U10, U11, U12, U13, U2, U3, U4, U5, U6, U7, U8, U9	Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Laboratorium	Metoda laboratoryjna, Metoda projektu, Demonstracje dźwiękowe i/lub video

Forma zajęć	Warunki zaliczenia zajęć
Laboratorium	Końcowa ocena jest wynikiem punktacji za zestaw programistycznych prac projektowych. Skala ocen: 1. bardzo dobry (bdb; 5,0) – od 90% punktów, 2. dobry plus (db plus; 4,5) – od 80% punktów, 3. dobry (db; 4,0) – od 70% punktów, 4. dostateczny plus (dst plus; 3,5) – od 60% punktów, 5. dostateczny (dst; 3,0) – od 50% punktów, 6. niedostateczny (ndst; 2,0) – poniżej 50% punktów.

Literatura

Obowiązkowa

1. M. Gonçalo, Elevating Game Experiences with Unreal Engine 5 - Second Edition: Bring your game ideas to life using the new Unreal Engine 5 and C++, Packt Publishing, 2022.
2. L. Mitchel, Game Development with Unreal Engine 5: Learn the Basics of Game Development in Unreal Engine 5 (English Edition), BPB Publications, 2022.
3. C. Chihming, Unreal Engine 5 RPG Development with C++ and Blueprint: Volume I: Basics, Combat, and VFX, Independently published, 2022.

Dodatkowa

1. H. Venter, W. Ogterol, Unreal Engine 5 Character Creation, Animation and Cinematics: Create custom 3D assets and bring them to life in Unreal Engine 5 using MetaHuman, Lumen, and Nanite, Packt Publishing, 2022.

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Laboratorium	60
Czytanie wskazanej literatury	20
Przygotowanie projektu	45
Łączny nakład pracy studenta	Liczba godzin 125
Liczba punktów ECTS	ECTS 5

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
INF_K3_U01_inz	Absolwent/ka potrafi zastosować wiedzę matematyczną do formułowania, modelowania, analizowania i rozwiązywania prostych zadań związanych z informatyką
INF_K3_U04_inz	Absolwent/ka potrafi opracować, przeanalizować, zaprojektować klasyczne algorytmy i systemy informatyczne
INF_K3_U05_inz	Absolwent/ka potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym
INF_K3_U06_inz	Absolwent/ka potrafi ocenić, na podstawowym poziomie, przydatność metod i narzędzi informatycznych oraz wybrać i zastosować właściwą metodę i narzędzia do typowych zadań informatycznych
INF_K3_U07_inz	Absolwent/ka potrafi zastosować wybrane metody wykorzystywane w wiodących kierunkach badań informatyki
INF_K3_W01	Absolwent/ka zna i rozumie zaawansowane pojęcia z działów matematyki służące do modelowania i rozwiązywania problemów w informatyce
INF_K3_W02	Absolwent/ka zna i rozumie zaawansowane pojęcia i problemy formujące kanon dyscypliny informatyka
INF_K3_W03	Absolwent/ka zna i rozumie narzędzia, technologie i urządzenia informatyczne właściwe dla wybranych obszarów zastosowań oraz podstawy ich działania
INF_K3_W04_inz	Absolwent/ka zna i rozumie zaawansowane pojęcia, konstrukcje i procesy związane z językami programowania i inżynierią programowania
INF_K3_W05	Absolwent/ka zna i rozumie zaawansowane pojęcia związane z algorytmami i strukturami danych
INF_K3_W10	Absolwent/ka zna i rozumie zaawansowane zagadnienia związane z grafiką komputerową