**Nazwa przedmiotu:**

Wytwarzanie gier komputerowych

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - podstawowe

**Kod przedmiotu:**

WGK

**Semestr nominalny:**

3 / rok ak. 2021/2022

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. liczba godzin kontaktowych – 62 godz., w tym
obecność na wykładach 30 godz.,
obecność na laboratorium 15 godz.,
obecność na spotkaniach projektowych 15 godz.,
obecność na konsultacjach 2 godz.
2. praca własna studenta – 56 godz., w tym
przygotowanie do sprawdzianów 8 godz.,
przygotowanie do laboratorium 8 godz.,
realizacja projektu 40 godz.
Łączny nakład pracy studenta wynosi 118 godz., co odpowiada 4 pkt. ECTS.

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

2,10 pkt. ECTS, co odpowiada 62 godz. kontaktowym

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

2,64 pkt. ECTS, co odpowiada 15 godz. ćwiczeń laboratoryjnych plus 8 godz. przygotowań do laboratorium plus 55 godzin realizacji projektu i spotkań projektowych

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 15h |
| Projekt: | 15h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Oczekuje się, iż uczestnicy przedmiotu będą posiadali podstawową wiedzę z grafiki komputerowej w zakresie metod modelowania i wizualizacji.

**Limit liczby studentów:**

30

**Cel przedmiotu:**

Nauka projektowania i wytwarzania gier komputerowych przy wykorzystaniu nowoczesnych, profesjonalnych środowisk programistycznych, określanych zwykle jako silniki gier.

**Treści kształcenia:**

Przedmiot poświęcony jest problematyce wytwarzania gier komputerowych przy wykorzystaniu nowoczesnych środowisk programistycznych, określanych często jako silniki gier. W skład każdego z takich środowisk wchodzi zwykle silnik właściwy, "wprawiający w ruch" zaprojektowany świat gry (w rozumieniu jego wizualizacji, zachodzących w nim interakcji fizycznych, zależności logicznych, itp.) oraz zbiór różnorakich narzędzi, które wspomagają proces tworzenia tego świata (różnego rodzaju edytory, interfejsy graficzne użytkownika, języki programowania). Na kolejnych jednostkach wykładowych omawiana jest budowa, używane algorytmy oraz narzędzia tworzące silnik gry na przykładzie nowoczesnego środowiska wytwarzania gier Unreal Engine 4 (UE4). Silnik ten jest dziś silnikiem wiodącym w przemyśle gier komputerowych i wykorzystywany jest przez wiele profesjonalnych studiów deweloperskich. Wykładowi towarzyszą seanse laboratoryjne, w trakcie których uczestnicy przedmiotu realizują ćwiczenia przy komputerach pod kierunkiem prowadzącego. Ćwiczenia te mają na celu sprawdzenie i wykorzystanie w praktyce wiedzy nabytej na wykładach. W skład przedmiotu wchodzi również projekt, w ramach którego uczestnicy mają za zadanie albo zaprojektowanie i zaimplementowanie własnego prostego silnika gry i następnie wykorzystanie go do utworzenia prostej gry, albo wytworzenie gry bardziej złożonej przy wykorzystaniu jednego silników dostępnych publicznie (w tym UE4).
Wykład:
Informacje ogólne o silniku gry: co to jest silnik gry, podstawowe składniki silnika gry, architektura i poszczególne moduły silnika gry.
Cykl produkcyjny tworzenia gry: od dokumentacji wstępnej projektu do prototypu świata gry i rozgrywki, prototypowanie świata gry na przykładzie UE4, siatki trójkątów i BSP, zastosowanie operatorów CSG, narzędzia modelowania geometrii BSP.
Prototypowanie świata gry: złożona struktura zasobów geometrycznych wykorzystywanych w grach, typy zasobów gry w UE4, różnice między zasobami, aktorami, a klasami; zarządzanie zasobami w UE4 – zadania Content Browsera;
Prototypowanie rozgrywki: wprowadzenie do systemu Blueprint, programowanie przy wykorzystaniu blueprintów jako programowanie zorientowane obiektowo; klasy, zdarzenia i funkcje, rodzaje klas blueprintowych; zmienne składowe, komponenty, kompozycja i dziedziczenie w blueprintach, edytor klas blueprintowych.
Prototypowanie rozgrywki (cd) oraz Programowanie w systemie Blueprint (cd.): elementy interaktywne świata gry, timeline, operacje na komponentach klas, kontrola przepływu w funkcjach blueprintowych, Construction Script jako analog konstruktora w OOP.
Komunikacja miedzy klasami oraz inne aspekty programowania w systemie Blueprint: komunikacja bezpośrednia i poprzez dynamiczne rzutowanie typu, przesyłanie zdarzeń (event dispatching), interfejsy i polimorfizm, referencje i polimorfizm, funkcje autonomiczne, makrodefinicje, biblioteki funkcji i makrodefinicji, tablice.
Ruch, interakcje między aktorami, kolizje: kategorie mobilności obiektów w grach komputerowych, rodzaje mobilności aktorów w UE4, przyczyny ruchu obiektów w grach komputerowych, wykrywanie kolizji, bryły kolizji w blueprintach, aktorzy-wyzwalacze (trigger volumes), zdarzenia: intersekcje i uderzenia, filtrowanie kolizji i kanały kolizji w UE4, rezultaty kolizji a zdarzenia, kolizje a śledzenie promieni i innych rodzajów prymitywów geometrycznych.
Interaktywne sterowani ruchem aktorów i kamer w UE4: transformacje aktorów, kamery jako autonomiczni aktorzy oraz komponenty, kontroler gracza (Player Controller), urządzenia sterujące, hierarchia przekazywania sygnału wejścia, Pionek (Pawn) oraz jego zawładnięcie przez kontroler gracza, klasy Pionka.
Animacja szkieletowa (1): ogólna idea animacji szkieletowej, szkielet, przekształcenia kości, skinning i mieszanie wierzchołków (vertex blending), animacje elementarne i złożone, przekierowanie animacji (retargetting), mieszanie animacji, przestrzenie mieszania i blueprinty animacji.
Animacja szkieletowa (2): rodzaje przestrzeni mieszania, wykorzystanie animacji szkieletowych w świecie gry, zastosowanie bluprintów animacji, graf zdarzeń i graf animacji, operacje mieszania w grafie animacji.
Animacja szkieletowa (3): idea maszyny stanu animacji, maszyna stanów animacji w UE4, programowanie stanów, przejść oraz przejść złożonych (conduit), animacja addytywna, zasób Aim Offset – jego tworzenie i wykorzystanie, łączenie aktorów – gniazdka.
Sztuczna inteligencja w grach (1): SI akademicka vs. SI w grach komputerowych, główne zagadnienia SI w grach, popularne metody implementacji SI w grach, drzewa decyzyjne, skończone maszyny stanów, hierarchiczne maszyny stanów, drzewa zachowań, podejmowanie decyzji w UE4 – zasoby: behavior tree i blackboard, trawersacja drzewa zachowań, węzły selector oraz sequence.
Sztuczna inteligencja w UE4 (2): dekoratory: idea i ich implementacja w UE4, dekoratory jako obserwatorzy – generowanie przerwań, serwisy, kontroler SI (AI Controller) i jego związek z klasą Pionka i drzewem zachowań, budowa Blackboarda i operacje zapisu/odczytu jego kluczy.
Sztuczna inteligencja w UE4 (3): węzły zadań (Task) i ich rola oraz możliwości, zadania predefiniowane, programowanie zadań, dekoratory predefiniowane, programowanie dektoratorów, programowanie serwisów, wpływ serwisów na kontrolę przepływu w drzewie zachowań, zagadnienia nawigacji w świecie gry (pathfinding), siatki nawigacji, komponenty symulowania zmysłów w UE4.
Laboratorium:
Na kolejnych seansach laboratoryjnych uczestnicy przedmiotu ćwiczą przy komputerach, pod kierunkiem prowadzącego, wykorzystanie w praktyce narzędzi i metod przedstawianych na wykładzie.
Projekt:
W ramach projektu uczestnicy przedmiotu, w zespołach dwuosobowych, albo implementują swój własny silnik gry i demonstrują jego działanie na przykładzie stworzonej przez siebie prostej gry, albo tworzą bardziej złożoną grę przy wykorzystaniu silnika istniejącego (w szczególności UE4).

**Metody oceny:**

Zajęcia laboratoryjne mają charakter szkoleniowy (ang. tutorials). Prowadzący omawia kolejne kroki w procesie tworzenia elementów gry komputerowej, przedstawiając na ekranie elementy interfejsu i na tym tle formułuje dla studentów na bieżąco niewielkie zadania laboratoryjne związane z tematem laboratorium.

**Egzamin:**

nie

**Literatura:**

D.H. Eberly, 3D Game Engine Design, 2nd ed. , Morgan Kaufmann, 2006.
J. Gregory, Game Engine Architecture, A K Peters/CRC Press, 2009.
https://docs.unrealengine.com
Brendel Sewell, Blueprints Visual Scripting for Unreal Engine, Packt Publishing 2015.
Aram Cookson, i inni, Unreal Engine 4 Game Development in 24 hours,SAMS, 2016.
Nicola Valcasara, Unreal Engine Game Development Bluprints, Packt Publishing, 2015.
Peter Newton, Unreal Engine 4 AI Programming Essentials, Packt Publishing, 2016.
Katax Emperore, Devin Sherry, Unreal Engine Physics Essentials, Packt Publishing 2015.

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103A-INIMU-MSP-WGK

**Uwagi:**

(-)

## Charakterystyki przedmiotowe