



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie gier

Przedmiot

Kierunek studiów

informatyka

Studia w zakresie (specjalność)

Zaawansowane technologie internetowe

Poziom studiów

drugiego stopnia

Forma studiów

niestacjonarne

Rok/semestr

1/2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

16

Ćwiczenia

Laboratoria

18

Projekty/seminaria

Inne (np. online)

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Paweł Wojciechowski

email: Pawel.Wojciechowski@cs.put.poznan.pl

tel. 61 6653031

Instytut Informatyki

ul Piotrowo 2, 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu programowania, grafiki trójwymiarowej, sieci komputerowych, sztucznej inteligencji oraz matematyki w zakresie działań na wektorach i macierzach. Powinien również posiadać umiejętność rozwiązywania



podstawowych problemów z zakresu algorytmiki, optymalizacji kodu programu oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.

Cel przedmiotu

Przekazanie studentom podstawowej wiedzy o problemach spotykanych przy programowaniu gier video, w zakresie sposobu działania silników grafiki oraz fizyki zarówno dwu, jaki i trójwymiarowych, działania skryptów sztucznej inteligencji i niskopoziomowego programowania efektów wizualnych, technik animacji oraz modelowania trójwymiarowego. Rozwijanie u studentów umiejętności przygotowania modelu trójwymiarowego danego obiektu wraz z jego animacją.

Przedmiotowe efekty uczenia się

Wiedza

1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie: funkcjonalności silników grafiki dwu i trójwymiarowej, sposobu opisu świata dla skryptów sztucznej inteligencji w grach, potoku przetwarzania grafiki trójwymiarowej.
2. ma szczegółową wiedzę w zakresie architektury i działania silników grafiki 3D zna podstawowe wymagania dotyczące symulacji fizyki i łączenia jej z wizualizacją.
3. zna podstawowe techniki animacji modeli 3D, zna przykładowe narzędzie stosowane do modelowaniu obiektów trójwymiarowych.
4. ma podstawową wiedzę o trendach w ewolucji trójwymiarowej grafiki komputerowej i kart graficznych

Umiejętności

1. potrafi pozyskiwać informacje z literatury, dokumentacji oraz forum dyskusyjnym dotyczącym silnika UnrealEngine 4 (w języku polskim i angielskim), integrować je, dokonywać ich interpretacji i stosować alternatywne rozwiązania do podanych na zajęciach.
2. potrafi przy formułowaniu i rozwiązywaniu zadań inżynierskich takich jak programowanie prostej gry, integrować wiedzę zarówno z różnych obszarów informatyki (np programowanie obiektowe, metody sztucznej inteligencji, programowanie sieciowe, przetwarzanie danych obiektów trójwymiarowych), jak i innych dyscyplin naukowych takich jak fizyka.
3. potrafi ocenić przydatność i możliwość wykorzystania poszczególnych wersji modułów silnika UnrealEngine 4.
4. potrafi ocenić przydatność silników grafiki 3D wspomagających programistę w procesie implementacji gry, w tym dostrzec celowość ich stosowania.

Kompetencje społeczne

rozumie, że w obszarze gier komputerowych wiedza i umiejętności bardzo szybko stają się przestarzałe, stąd istnieje potrzeba ciągłego uczenia się



Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Na wykładzie odbywa się turniej skryptów sztucznej inteligencji. Wykład kończy się egzaminem

Laboratoria podzielone są na poszczególne etapy związane z konkretnymi zagadnieniami. Każdy z 3-4 etapów wymaga realizacji stosunkowo krótkiego zadania, które należy przedstawić prowadzącemu. Zadania te nie są oceniane. Dodatkowo, każdy student realizuje projekt zaliczeniowy, którym jest gra komputerowa zaimplementowana w wybranej technologii

Treści programowe

Wykład:

Problematyka sztucznej inteligencji w grach realizowana jest w oparciu o projekt *evaLUation* przygotowany w ramach zajęć w latach ubiegłych. W ramach wykładu, przedstawione są podstawowe założenia środowiska, omówiony problem interakcji skryptu z otoczeniem i sposób opisu otoczenia widzianego z poziomu skryptu.

Wprowadzenie do silnika gier UnrealEngine. Interakcja obiektów gry ze środowiskiem. Omówienie rodzajów obiektów występujących w grze i ich właściwości. Wprowadzenie właściwości fizycznych wybranych obiektów w tym sposoby ograniczania symulacji do dwóch wymiarów.

Zasady modelowania 3D na przykładzie aplikacji Blender. Zapoznanie się z narzędziem, podstawowe własności modeli, metody teksturowania, rodzaje oświetlenia, mapowanie nierówności. Rodzaje animacji w grafice 3D na przykładzie programu Blender. Pojęcia klatek kluczowych, rodzaje interpolacji, animacja szkieletowa oraz deformacje siatki obiektów (ang. Shape Keys). Zasady eksportu modeli i animacji do silnika UnrealEngine.

Wykorzystanie modelu w procesie budowy gry 3D i importu standardowej postaci.

Właściwości silnika grafiki 3D Unreal Engine, podstawowe przekształcenia obiektów, import modeli z programu Blender, animacje, system tworzenia terenu. Wykorzystanie fizyki. Przedstawienie problemów i możliwości badanych rozwiązań.

Omówienie potoku grafiki 3D, ewolucja kart graficznych, języki ich programowania, wprowadzenie podstawowych pojęć: pixel shader, vertex shader, geometry shader, cieniowanie wierzchołków i pikseli, mapowanie nierówności.

Zajęcia laboratoryjne podzielone są na bloki tematyczne:

Sztuczna inteligencja, w ramach której studenci zapoznają się ze środowiskiem *evaLUation* służącym do nauki pisania skryptów walczących w języku LUA. Każdy ze studentów przygotowuje drużynę składającą się z trzech postaci, które następnie biorą udział w turnieju.



Wprowadzenie do silnika UnrealEngine - wykorzystanie podstawowych narzędzi, blueprints na przykładzie implementacji prostej gry platformowej z wykorzystaniem modelu przygotowanego wraz z teksturami w środowisku blender.

Omówienie zagadnień związanych z modułem fizyki w środowisku UE4 - podstawowe mechanizmy kolizji, ruchu oraz zależności między obiektami. Obsługa podstawowych zdarzeń generowanych w trakcie zderzeń obiektów. Wprowadzenie do pojęcia materiału fizycznego.

Omówienie materiałów i wprowadzenie do systemu cząstek. Wykorzystanie elementów interfejsu użytkownika (GUI).

Metody dydaktyczne

1. wykład: prezentacje multimedialna, wraz z demonstracją następującego oprogramowania: evalUAtion, Blender, UnrealEnigne4
2. ćwiczenia laboratoryjne: rozwiązywanie zadań obejmujących wykorzystanie wybranych technologii, turniej walczących skryptów

Literatura

Podstawowa

1. Perełki programowania gier t. 1, 2 i 3, DeLoura M., Helion, 2002
2. Fizyka dla programistów gier, Bourg, D.M., O'Reilly; Associates, 2003
3. Mathematics for 3D Game Programming and Computer Graphics, Third Edition, Lengyel, E., Course Technology PTR, 2011
4. Programming Game AI by Example, Buckland, M., Jones Bartlett Publishers, 2004
5. Game Coding Complete 4 edition, McShaffry, M., Graham, D., Course Technology PTR, 2012
6. Język Cg. Programowanie grafiki w czasie rzeczywistym, Fernando, R., Kilgard, M.J., Helion, 2003
7. Blender. Kompendium, Kuklo, K., Kolmaga, J., Helion, 2007
8. Animacja komputerowa Algorytmy i techniki, Parent, R., PWN, 2011

Uzupełniająca

1. GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics, Fernando, R. (Series Editor), Addison Wesley Professional, 2004
2. GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, Pharr, M., Fernando, R. (Series Editor), Addison Wesley Professional, 2005



Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	101	4,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1,5
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, napisanie skryptów sztucznej inteligencji, utworzenie modelu 3D i jego animacji szkieletowej, wykonanie projektu zaliczeniowego) ¹	67	2,5

¹ niepotrzebne skreślić lub dopisać inne czynności