



Використання технології ігрових застосувань Unity для радіотехнічних систем

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти Перший (бакалаврський)

Галузь знань	17 — Електроніка, автоматизація та електронні комунікації
Спеціальність	172 – Електронні комунікації та радіотехніка
Освітня програма	Інтелектуальні технології радіоелектронної техніки Інформаційна та комунікаційна радіоінженерія Радіотехнічні комп'ютеризовані системи
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)
Рік підготовки, семестр	3 рік, 5 семестр
Обсяг дисципліни	120 годин (18 годин – Лекції, 36 годин – Лабораторні, 66 годин – СРС)
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент Катін Павло Юрійович, ka72tin@gmail.com , моб. +38(098)202-08-11 Лабораторні: к.т.н., доцент Катін Павло Юрійович
Розміщення документу	

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Опис дисципліни. Основною метою дисципліни “Розробка ігрових застосувань” є надання системних знань про:

- створення прототипів радіотехнічних систем у вигляді емуляції на основі ігрового рушія;
- основи мови програмування для розробки скриптів;
- основи інтерфейсу ігрового рушія та ітерованого середовища розробки ігор;
- базові підходи і навички для створення ігор та ігрових сцен у двовимірному і просторовому варіантах;
- види, технології, програмні платформи і ітеровані середовища розробки ігрових застосувань на прикладах Unity;
- організацію і склад групи розробників, способи монетизації, споріднені технології ігрових застосувань;
- програмну архітектуру і сучасні технології ігрових застосувань (ІЗ);
- про мінімально-потрібний обсяг знань мови програмування для розробки ІЗ;
- основи програмної архітектури програмного “двигуна” ІЗ;
- структуру і загальні принципи побудови дизайну ІЗ;
- елементи логіки ІЗ та типових патернів програмування.

Крім того дисципліна надає навички розробки ІЗ з використанням обраної технології ІЗ і ігрового рушія.

Силабус навчальної дисципліни передбачає виконання навчальних завдань та практик програмування, що потрібні студентам для досягнення результатів навчання з урахуванням особливого режиму.

Присутній стандартний підхід з навчання програмуванню, від простого до складного, що спрощує процес подальшого навчання. Курс призначений для студентів, що не володіють мовою програмування для розробки ігрових скриптів.

Фінальним завданням є розробка прототипу ІЗ. Для виконання цього завдання студенти використовують теоретичні знання та застосовують практичні навички, отримані під час всього лекційного курсу і лабораторних робіт.

Предмет навчальної дисципліни: процес розробки ІЗ та прототипів радіотехнічних систем на основі ігрового рушія.

Міждисциплінарні зв'язки: Попередні дисципліни: Основи програмування; Дискретна математика, Фізика.

Дисципліни, що забезпечуються: бакалаврський проект, магістерська робота (у питаннях вибору, розробки і обґрунтування рішень).

Мета навчальної дисципліни. Підготовка розробників та менеджерів у галузі ІЗ з метою створення прототипів радіотехнічних систем на основі ігрового рушія.

Результатами навчання є

Знання:

- варіантів створення емуляції радіотехнічних систем на основі ігрового рушія;
- видів ігрових застосувань, їх особливості, історію, рейтинги, комерційний рівень, способи монетизації;
- основи математичного супроводження і елементів штучного інтелекту розробки ІЗ;
- сучасні технології ігрових застосувань, переваги і недоліки різних технологій;
- основи мови програмування, що використовуються як скрипти для розробки ігрових застосувань, переваги і недоліки різних мов програмування;
- досконале знання однієї з мов програмування для розробки ІЗ, типові патерни, що використовуються у галузі ІЗ;
- основи програмної архітектури програмного “двигуна” ІЗ, різні види “двигунів”, їх переваги та недоліки, досконале знання одного з них;
- механізми побудови графічного інтерфейсу і дизайну;
- основи логіки комп'ютерної гри;
- основи технології підготовки до релізу.

Уміння:

- розробляти прототип емуляції радіотехнічної системи у команді під керівництвом досвідченого менеджера і розробника (рівень джуніор);
- самостійно розробляти прототип комп'ютерної гри з використанням однієї з обраних технологій;
- розробляти дизайн, керувати розробкою дизайну комп'ютерної гри;
- здійснювати монетизацію рішень, тестувати і підтримувати розроблене ІЗ.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити: Основи програмування; знати одну з мов програмування, що є основою скриптів ігрового рушія (C#, C++, C); базові знання дискретної математики та аналітичної геометрії, знання англійської мови відповідно до програми, знання фізики.

Постреквізити: Отримані під час засвоєння дисципліни (залікового модуля) теоретичні знання та практичні навички необхідні для розробки бакалаврського проекту.

3. Зміст навчальної дисципліни

Лекційні заняття.

Тема 1. Основи комп'ютерного рушія та види комп'ютерних ігор.

Лекція 1. Види комп'ютерних ігор і технології їх розробки на прикладі Unity (Unreal 5).

Види комп'ютерних ігор. Питання особистої безпеки під час взаємодії із ігровими програмами. Ігрові рушії і технології розробки ігрових застосунків. Підготовка репозиторію ігрового проекту.

Тема 2. Створення концепції гри та проведення бізнес аналізу

Лекція 2. Основи планування, організації і аналітики розробки ігрового застосунку та систем емуляції радіотехнічних систем.

Жанри комп'ютерних ігор. Напрямки 3D, 2D. Створення систем емуляції з використанням ігрового рушія. Розбиття на проектні команди і визначення напрямку гри. Основи аналітики.

Розробка ігрової концепції. Декомпозиція на механіки. Опис і формалізація Core Gameplay та Meta Gameplay. Процес створення проекту. Підключення інструментарію та знайомство з Asset Store. Організація файлів проекту і основи скриптів.

Тема 3. Основи інтегрованого середовища розробки ігрових застосувань та програмування скриптів.

Лекція 3. Базові елементи інтегрованого середовища розробки ігрових застосувань та використання типів даних.

Основи інтегрованого середовища розробки скриптів. Базовий патерн скрипту ігрового рушія Unity (Unreal 5). Зв'язок елементів інтегрованого середовища ігрового застосування та скрипту гри на прикладі термінального виводу. Відмінність технології версії 2022.3.62f LTS (та ранніх версій) від версії 6000.2.2f1 LTS. Основи змінних (полів), типів даних, базове поняття про методи. Основи потоку управління.

Тема 4. Програмна архітектура скриптів ігрового рушія.

Лекція 4. Основи базової архітектури скриптів на прикладі методів ігрового рушія.

Управління ігровим персонажем. Основи проекту ігрового рушія і програмної архітектури скриптів на прикладі управління ігровим персонажем Unity (Unreal 5) проста сцена 3D (2D).

Налагодження вихідного коду скрипту з використанням виводу повідомлення на консоль. Оголошення змінних різних типів та їх ініціалізація. Використання модифікаторів доступу, особливості модифікаторів доступу у контексті використання в ігровому рушії. Розуміння області

бачення змінних. Основи оголошення, визначення та виклику методу у рамках одного класу. Перелік і довідникова інформація про типові методи базового патерну ігрового рушія.

Тема 5. Програмна складова комп'ютерної гри на базі рушія (Unity, Unreal Engine 5) на базі простої сцени 3D (2D).

Лекція 5. Основи управління потоком у скрипті гри та оператори утворення циклів.

Оператори потоку управління в програмі (скрипті ігрового рушія). Оператори організації циклів. Додаткові елементи для операторів циклу. Контейнери сутностей та колекції.

Тема 6. Основи об'єктно-орієнтованого програмування у системи скриптів ігрового рушія.

Лекція 6. Класи ігрового рушія.

Оголошення, опис, створення екземпляру класу та структури. Використання методів, властивостей та інших членів класу. Використання полів структури. Загальні відомості, синтаксис посилань. Основні принципи об'єктно-орієнтованого програмування. Особливості ООП у ігрових рушіях.

Тема 7. Використання ігрового рушія для створення складових елементів гри або емулятора радіотехнічної системи.

Лекція 7. Створення рівня та сцени гри (емуляції) у ігровому рушії.

Основи ігрового дизайну та емуляції. Технології створення рівня гри. Ігрові об'єкти та префаби. Програмне управління емуляцією освітлення. Анімація ігрового персонажа та навколишнього середовища.

Тема 8. Програмне управління ігровим та неігровими персонажами у сцені комп'ютерної гри.

Лекція 8. Сцена з управлінням ігровими персонажами.

Програмне управління моделлю ігрового персонажа у контексті руху та обертання. Обробка активності гравця через введення з клавіатури, миші та інших пристроях вводу на рівні скрипту. Програмування вигляду гри з боку гравця (програмована поведінка відеокамери). Моделювання фізики у ігровому рушії. Базові колайдери та тригери гри.

Тема 9. Механіка гри та штучний інтелект неігрових персонажів.

Лекція 9. Досконалі складові комп'ютерної гри та штучний інтелект.

Програмування стрибків. Використання шарів у системі гри. Створення екземплярів префабів та об'єктів. Програмна складова менеджера гри. Інтерфейс користувача комп'ютерної

гри. Система навігації ігрового рушія та навігаційні мережі. Штучний інтелект неігрових персонажів.

Лабораторні заняття.

Назва лабораторної роботи
Лабораторна робота 1. * Дослідження базового патерну ігрового рушія Unity на прикладі двовимірної технології
Лабораторна робота 2. * Дослідження базового патерну ігрового рушія Unity на прикладі тривимірного ігрового застосунку
Лабораторна робота 3. Дослідження типових патернів у скриптах ігрового застосунку
Лабораторна робота 4. Документування концепції гри
Лабораторна робота 5. Базові складові ігрового застосунку
Лабораторна робота 6. Бізнесова складова ігрового застосунку
Лабораторна робота 7. Мережеві елементи ігрового застосунку
Лабораторна робота 8. Основи штучного інтелекту гри і інтерфейсу користувача гри

4. Навчальні матеріали та ресурси

Базова література

1. Learning C# Programming with Unity 3D by Alex Okita.
2. Unity in Action: Multiplatform Game Development in C# with Unity 5 by Joe Hocking.
3. Introduction to Game Programming: Using C# and Unity 3D by Vahe Karamian.
4. Learning C# by Developing Games with Unity 3D Beginner's GuideSep by Terry Norton.
5. C# Game Programming Cookbook for Unity 3D by Jeff W. Murray.

Допоміжна література

1. Unity 3D UI Essentials by Simon Jackson.
2. Game Programming Patterns by Robert Nystrom.
3. Creating E-Learning Games with Unity by David Horachek.
4. Learn Unity for 2D Game Development by Alan Thorn.
5. Unity for Architectural Visualization by Stefan Boeykens.
6. Вилучено
7. Вилучено
8. <https://unity.com>
9. <https://www.unrealengine.com/en-US/>
10. <https://blog.studica.com/how-to-setup-github-with-unity-step-by-step-instructions>
11. https://uk.wikipedia.org/wiki/Жанри_відеоігор

12. <https://dtf.ru/gamedev/5375-opisanie-ciklov-igry-primery-i-sovety>
13. <https://www.epravda.com.ua/rus/publications/2018/09/11/640413>
14. <https://www.epravda.com.ua/publications/2021/11/26/680153/>
15. <https://docs.unity3d.com/ru/530/Manual/ExecutionOrder.html>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття (18 годин,)

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на літературу та завдання на СРС)
1	Тема 1. Основи комп'ютерного рушія та складових елементів гри
	Вивчення графічного інтерфейсу ігрового рушія. Підготовка репозиторію та флоу, таск менеджера. Створення проекту. Підключення інструментарію та знайомство з Asset Store. Організація файлів проекту. *Реалізація 1 сцени на прикладі 3D гри на основі готового скріншоту, що надається компоненти Rigidbody. Документація Unity приклад Rigidbody. Без скриптова.
2	Тема 2. Підготовка і формалізація основ концепції гри та основи бізнес аналізу.
	1. Жанри комп'ютерних ігор. Напрямки 3D, 2D. Розбиття на проектні команди і визначення напрямку гри. Напрямок 3D, 2D. 2. Основи аналітики. Розробка ігрової концепції. Декомпозиція на механіки. 3. Опис і формалізація Core Gameplay та Meta Gameplay. 4. Базові питання моделі і вирішення питань монетизації. 5. Основи програмування.
3	Тема 3. Математичні моделі рушія Unity (або Unreal Engine за вибором) і реалізація фізичних моделей.
	Життєвий цикл моно біхев. Зв'язки прокидання скриптів на пряму, як створення зв'язків. Основи створення та налаштування базових елементів системи управління Створення та налаштування сцени персонажа гри і налаштування програмних сутностей та інших об'єктів у системі 3D. Основи скриптового управління ігровим персонажем. Основи управління, завантаження сцен, управління графічними і звуковими сутностями з боку гравця за рахунок використання програмних скриптів на основі мови C#. Граф примітив та зображення.

	Вибір і створення основних скриптів архітектурного патерну. MVC, ECS.
4	Тема 4. Математичні моделі рушія Unity (або Unreal Engine за вибором) і реалізація фізичних моделей.
	Ігровий простір. Проста сцена з платформою і персонажем гри (на прикладі примітиву, або картинки в 2d). Просторові математичні моделі рушія Unity на основі лінійної алгебри і аналітичної геометрії. Векторна математика. Фізичні моделі у Unity Physics. Програмне налаштування і управління параметрами фізичних моделей. Реалізація моделей руху, поворотів, балістичних розрахунків. Практична реалізація програмної архітектури на основі типових патернів (шаблонів).
5	Тема 5. Реалізація ігрових сутностей на основі математичного і фізичного моделювання у ігровому рушії Unity (або Unreal Engine за вибором).
	Одинична сцена, з персонажа гри (на прикладі примітиву 3d, або картинки в 2d). Логіка гри. Ігрова механіка. Подія. Патерн стратегія. Створення систем та модулів ігрових сутностей Unity на основі лінійної алгебри і аналітичної геометрії та фізичних моделей. Приклади реалізації ігрової механіки. Реалізація архітектури гри. Практичні приклади на основі ігрових рішень.
6	Тема 6. Досконалі налаштування зовнішнього вигляду ігрового персонажа у системі анімації 3D (2D) і мета-механіка гри.
	Патерн Стейт, стейт машина. Графічна, програмна реалізація і типові налаштування програмних сутностей персонажа гри у системі 3D. Скриптове управління ігровим персонажем. Графічна, програмна реалізація і типові налаштування програмних сутностей NPC (Non-Player Character) та інших об'єктів. Особливості скриптового управління. Особливості анімації, Animator Controller. Скриптові анімації на прикладі двигуна DOTween. Основи створення мета-механіки гри. Прокачування, рівні, характеристики. Збереження гри. *Досконала робота системи контролю версій.
7	Тема 7. Досконалі налаштування ігрового простору у системі 3D (2D) і тестування

	гри.
	Робота з Terrain та налаштування оточення. Розміщення ігрового оточення. Налаштування камери ігрового персонажа у системі 3D (2D). Потреба і особливості використання множин відеокамер. Налаштування освітлення, особливості реалізації освітлення, “запікання” світла. Налаштування камери 2 D. Впровадження тестування у проект. Складання тест плану. Налагодження процесів тестування. Планування тестування модулів. Техніки тест дизайну. Складання тест-кейсів та чек-листів. Безпосереднє тестування модуля. Тестування перед релізом та після. Вектори розвитку тестувальника: автоматизація тестування, тестування продуктивності, тест на зламування (ethical hacking).
8	Тема 8. Художнє оформлення ігрового простору у системі 3D і створення музичної емоційної складової і монетизація гри
	Робота з Particle System. Налаштування шейдера і матеріалів 3D (2D). Пост процесинг. Технологія додавання до гри звукових ефектів і аудіомікшерів. Впровадження внутрішньоігрових покупок та підписок. Впровадження реклами. Вбудовування аналітики. Метрики та аналіз гри.
9	Тема 9. Основи штучного інтелекту гри і інтерфейсу користувача гри.
	Основи роботи з інтерфейсом. Створення основних вікон. Налаштування елементів інтерфейсу. Зв'язок інтерфейсу та механік. Реалізація основ штучного інтелекту персонажа гри у системі 3D. Стейт машина. Реалізація основ штучного інтелекту програмних сутностей NPC (Non-Player Character) та інших об'єктів. Особливості скриптового управління. Збереження ігрових релізів. Основи роботи з сервісами App Store, Play market, Steam. Організація роботи з ревью маркету.

Лабораторні заняття (Дозволяються по узгодженню з викладачем захищати проекти, що отримані (розроблені) на курсах, при виконанні ЛР1-2)

№	Назва лабораторної роботи	Кількість ауд. годин
1	Лабораторна робота 1. * Дослідження базового патерну ігрового рушія Unity на прикладі двовимірної технології	4
2	Лабораторна робота 2. * Дослідження базового патерну ігрового рушія Unity на прикладі тривимірного ігрового застосунку	4
3	Лабораторна робота 3. Дослідження типових патернів у скриптах ігрового застосунку	4
4	Лабораторна робота 4. Документування концепції гри	4
5	Лабораторна робота 5. Базові складові ігрового застосунку	4
6	Лабораторна робота 6. Бізнесова складова ігрового застосунку	4
7	Лабораторна робота 7. Мережеві елементи ігрового застосунку	6
8	Лабораторна робота 8. Основи штучного інтелекту гри і інтерфейсу користувача гри	6

6. Самостійна робота студента/аспіранта (66 години)

Дисципліна ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

№	Напрямок самостійної підготовки	Кількість годин	Література
1	Підготовка до лекції 1	1	1-5
2	Підготовка до лабораторної роботи 1	1,5	1-5
3	Підготовка до лекції 1	1	1-5
4	Підготовка до лекції 2	1	1-5
5	Підготовка до лабораторної роботи 2	1,5	1-5
6	Підготовка до лекції 2	1	1-5
7	Підготовка до лекції 3	1	1-5
8	Підготовка до лабораторної роботи 3 (part 1)	1,5	1-5
9	Підготовка до лекції 3	1	1-5
10	Підготовка до лекції 4	1	1-5
11	Підготовка до лабораторної роботи 3 (part 2)	1,5	1-5
12	Підготовка до лекції 4	1	1-5
13	Підготовка до лекції 5	1	1-5
14	Підготовка до лабораторної роботи 4 (part 1)	1,5	1-5

15	Підготовка до лекції 5	1	1-5
16	Підготовка до лекції 6	1	1-5
17	Підготовка до лабораторної роботи 4 (part 2)	1,5	
18	Підготовка до лекції 7	1	
19	Підготовка до лекції 7	1	
20	Підготовка до лабораторної роботи 5	1,5	
21	Підготовка до лекції 8	1	1-5
22	Підготовка до лекції 8	1	1-5
23	Підготовка до лабораторної роботи 6 (part 1)	1,5	1-5
24	Підготовка до лекції 9	1	1-5
25	Підготовка до лекції 9	1	1-5
26	Підготовка до лабораторної роботи 7, 8 (part 2)	1,5	1-5
27	Preparation for the midterm test	4	1-5
28	Preparation for the exam	30	1-5
29	Getting started with MVS	2	1
30	Install MVS	1	1
31	Essential C# Features for web developing. Stage 1.1	1	1
32	Essential C# Features for web developing. Stage 1.2	1	1
33	Essential C# Features for web developing. Stage 2.1	1,5	1
34	Essential C# Features for web developing. Stage 2.2	1	1
35	Essential C# Features for web developing. Stage 3.1	1	1
36	Essential C# Features for web developing. Stage 3.2	2	1
37	Essential C# Features for web developing. Stage 4.1	2	1
38	Essential C# Features for web developing. Stage 4.2	1	1
39	Essential C# Features for web developing. Stage 5	1	1
40	Essential C# Features for web developing. Stage 6	2	1

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних занять є обов'язковим. В умовах особливого стану питання щодо відвідування лекційних занять і інших аспектів політики може бути змінені.

Відвідування занять комп'ютерного практикуму може бути епізодичним та за потреби консультації/захисту робіт лабораторних робіт.

Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.

Дотримання політики академічної доброчесності.

Правила захисту лабораторних робіт: роботи повинні бути зроблені відповідно до поставлених задач та згідно з варіантом.

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Протягом семестру студенти виконують 8 лабораторних робіт. Максимальна кількість балів за кожен: 10 балів.

Бали нараховуються за:

- якість виконання лабораторних робіт: 0-5 балів;
- відповідь під час захисту лабораторних робіт: 0-3 балів;
- своєчасне представлення роботи до захисту: 0-2 балів.

Критерії оцінювання якості виконання:

- 5 балів – робота виконана якісно, в повному обсязі;
- 4 бали – робота виконана якісно, в повному обсязі, але має недоліки;
- 3 бали – робота виконана в повному обсязі, але містить незначні помилки;
- 2 бали – робота виконана в повному обсязі, але містить суттєві помилки;
- 0 балів – робота виконана не в повному обсязі.

Критерії оцінювання відповіді:

- 3 бали – відповідь повна, добре аргументована;
- 2 бали – відповідь вірна, але має недоліки або незначні помилки;
- 1 бал – у відповіді є суттєві помилки;
- 0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання своєчасності представлення роботи до захисту:

- 2 бали – робота представлена до захисту не пізніше вказаного терміну;
- 0 балів – робота представлена до захисту пізніше вказаного терміну.

Максимальна кількість балів за виконання та захист лабораторних робіт:

$10 \text{ балів} \times 8 \text{ л.р.} = 80 \text{ балів.}$

Протягом семестру на лекціях відбуваються **опитування за темою поточного заняття**. Максимальна кількість балів за всі опитування: 3 бали. Кількість **опитування за темою поточного заняття** для одного студента є необмеженою.

Завдання на **модульну контрольну роботу** складається з 1 теоретичних та 1 практичних запитань. Відповідь оцінюється максимально 20 балами.

Критерії оцінювання кожного запитання контрольної роботи:

- 9-10 балів – відповідь вірна, повна, добре аргументована;
- 7-8 балів – відповідь вірна, розгорнута, але не дуже добре аргументована;
- 5-6 балів – в цілому відповідь вірна, але має недоліки;
- 3-4 балів – у відповіді є незначні помилки;
- 1-2 бали – у відповіді є суттєві помилки;
- 0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:

$10 \text{ балів} \times 2 \text{ запитань} = 20 \text{ балів.}$

Рейтингова шкала з дисципліни дорівнює:

$R = R_C = 80 \text{ балів} + 20 \text{ балів} = 100 \text{ балів.}$

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог програми.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 15 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 20 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: залік

Умови допуску до семестрового контролю:

При семестровому рейтингу (R_c) не менше 60 балів та зарахуванні усіх робіт комп'ютерного практикуму, студент отримує залік «автоматом» відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою). В іншому разі він має виконувати залікову контрольну роботу.

Необхідною умовою допуску до залікової контрольної роботи є виконання і захист лабораторних робіт.

Якщо студент не погоджується з оцінкою «автоматом», то може спробувати підвищити свою оцінку шляхом написання залікової контрольної роботи, при цьому його бали, отримані за семестр, зберігаються, а з двох отриманих студентом оцінок виставляється краща («м'яка» система оцінювання).

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Для роботи можуть бути використані різні операційні системи за бажанням студента і по узгодженню з викладачем.

У випадку бажання студента використовувати у комп'ютерному практикумі інші мови програмування або технології вони можуть бути включені як додатковий елемент у базову архітектуру розподіленого веб-застосунку.

Робочу програму навчальної дисципліни (силабус):

Складено доцент кафедри, к.т.н., доцент, Катін П.Ю.

Ухвалено кафедрою РТС_ (протокол № 06/2024_ від _ 27.06. 2024 р.)

Погоджено Методичною комісією радіотехнічного факультету (протокол № _ 06/2024 _ від 28.06.2024 р.)

Програма підготовки до залікової співбесіди.

1. Основні елементи середовища розробки Unity3d.
2. Текстури і матеріали. Управління редактором ландшафтів.
3. Методика і практика створення текстур.
4. Методика і практика додавання нових текстур в проект.
5. Основи створення ландшафту на Terrain, навести приклади.
6. Технологія розміщення гравця на ландшафті.
7. Приклад створення скриптів.
8. Застосування скриптів до ігрового персонажу.
9. Застосування скриптів до елементів сцени.
10. Патерни скриптів у ігрових застосуваннях.
11. Архітектура і приклад використання у ігрових застосуваннях патерну Command.
12. Патерни поведінки у контексті ігрових застосувань. Архітектура і приклад використання у ігрових застосуваннях Bytecode.
13. Архітектура і приклад використання у ігрових застосуваннях патерну Flyweight.
14. Патерни поведінки у контексті ігрових застосувань. Архітектура і приклад використання у ігрових застосуваннях Subclass Sandbox.
15. Архітектура і приклад використання у ігрових застосуваннях патерну Prototype.
16. Патерни поведінки у контексті ігрових застосувань. Архітектура і приклад використання у ігрових застосуваннях Type Object.
17. Архітектура і приклад використання у ігрових застосуваннях патерну Singleton.
18. Патерни поведінки у контексті ігрових застосувань. Архітектура і приклад використання у ігрових застосуваннях Decoupling Patterns.
19. Обґрунтування використання у ігрових застосуваннях патерну State.
20. Архітектура і приклад використання у ігрових застосуваннях патерну State.
21. Структура скриптів патерну Game Loop.
22. Структура скриптів патерну Double Buffer.
23. Структура скриптів патерну Update Method.
24. Приклади елементарних скриптів руху у Unity3d.
25. Створення асертів і використання готових асертів у Asset store.
26. Методи розробки і тестування скриптів для ігрового персонажу. Тригери.
27. Особливості роботи з CharacterController.
28. Інтерфейс користувача і створення меню гри.
29. Анімація. Використання готових анімацій і створення власних анімацій.
30. Основи роботи з 2D іграми. Технологія створення 2D ігор.

31. Основи роботи зі спрайтами в Sprite Editor.

32. Жанри комп'ютерних ігор. Психологія комп'ютерних ігор. Аналіз культових ігор.