

Міністерство освіти і науки України
Національний технічний університет України
Київський політехнічний інститут імені Ігоря Сікорського

Катін П. Ю.

**РОЗРОБКА ІГРОВИХ ЗАСТОСУВАНЬ
(ПРОЕКТ)**

Лабораторний практикум
Електронне мережеве видання

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра першого року навчання
121 «Інженерія програмного забезпечення» факультету прикладної математики та
126 «Інформаційні системи та технології».*

Київ
КПІ ім. Ігоря Сікорського
2022

УДК 004.2(076.5) К29

К 29

Рекомендовано Методичною радою КПП ім. Ігоря Сікорського

(Протокол №

== від ==.==.====)

Рецензенти

канд. техн. наук, доц. кафедри обчислювальної техніки, ФІОТ, Національний
технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

, канд. техн. наук, доц.,

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Катін П. Ю.

К 29

Розробка ігрових застосувань: лаб. практикум [Електронний ресурс]: навч. посібн. для бакалаврів четвертого року навчання. Спеціальність 121 «Інженерія програмного забезпечення», 126 «Інформаційні системи та технології»/ П. Ю. Катін. – Електронні текстові дані (1 файл: 1,124 Мбайт). – Київ: КПП ім. Ігоря Сікорського, 2022. – 180 с.

Уміщено завдання на лабораторні роботи, приклади їх розв'язання та порядок оформлення результатів лабораторних робіт у контексті вивчення інфраструктури програмного забезпечення. Надано прототипи (шаблони) коду для використання у лабораторних роботах та додатковий інформаційний матеріал.

Для студентів всіх форм навчання бакалаврату, які навчаються за спеціальністю 121 «Інженерія програмного забезпечення» факультету прикладної математики та інших факультетів з цією спеціальністю та 126 «Інформаційні системи та технології».

УДК 004.2(076.5)

ЗМІСТ

Передмова

Інтегроване середовище для лабораторних робіт

Лабораторна робота 1. Створення проекту і репозиторію для контролю версій

1.1 Завдання

1.2 Теоретичні відомості

1.3 Програма співбесіди до лабораторної роботи № 1

Лабораторна робота 2. *Створення проекту комп'ютерної гри і дослідження базових елементів графічного інтерфейсу integrated develop environment (IDE) і рушія Unity на базі 2D гри.

2.1 Завдання

2.2 Теоретичні відомості

2.3 Програма співбесіди до лабораторної роботи № 1

ПЕРЕДМОВА

Навчальний посібник розроблений для освітніх програм 121 спеціальності: «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем», «Системне програмування та спеціалізовані комп'ютерні системи», та інших споріднених освітніх програм.

Для спеціальності 126 даний посібник може бути застосований для наступних освітніх програм: «Інтегровані інформаційні системи», «Інформаційні управляючі системи та технології», «Інформаційне забезпечення робототехнічних систем».

Виконання лабораторних робіт у даному курсі дозволяється на різних інтегрованих середовищах розробки ((ICP або IDE) Integrated development environment) і різних ігрових рушіях, що забезпечують зручний інструментарій для розробки програмного забезпечення. У якості основних ігрових рушіїв запропоновано обрати 2 найбільш відомих ігрових рушія:

Unity 3D, Microsoft Visual Studio, мова програмування C#;

Зважаючи на доволі широкий спектр технологій і видів ігрових застосувань і творчу складову дослідження, виконання послідовності лабораторних робіт у даному навчальному посібнику пропонується як базовий варіант. Деякі лабораторні роботи можуть бути поєднані у єдине ціле, як частина або складова проекту.

Захист лабораторних робіт відбувається шляхом співбесіди з демонстрацією робочого столу персонального комп'ютера і розробленого проекту комп'ютерної гри на базі ігрового рушія та IDE, що підготовлений до захисту. Під час захисту проекту студент демонструє володіння ігровим рушієм, знання мови синтаксису скриптів та особливостей їх інтеграції у проект комп'ютерної гри основи математичного моделювання. Крім того студенти мають бути підготовлені до відповідей на запитання, що продемонстровані у програмі, яка наведена у Додатку 1.

Результатом лабораторних робіт має стати діючий прототип комп'ютерної гри 3D (або 2D за вибором), вихідний код і проект, що розташовані у розподіленій системі контролю версій с комітами, що показані вище.

Інтегроване середовище для лабораторних робіт

Основним IDE, мовами програмування і базовими технологіями для курсу лабораторних робіт вибрані:

Unity 3D, Microsoft Visual Studio, мова програмування C#, технологія;

Виконання курсу лабораторних робіт дозволяється також і у інших IDE. У цьому випадку студент має обґрунтовано показати основні елементи програмної інфраструктури, акцентувати увагу саме на ці питання під час захисту ЛР, забезпечити відповідне документування цих технологій.

ЛАБОРАТОРНА РОБОТА № 1

Дослідження базового патерну ігрового рушія Unity на прикладі двовимірної технології

1.1 Завдання

Мета роботи: полягає у набутті знань, умінь та навичок з технології розроблення основ проекту з використанням обраної мови програмування у обраній парадигмі.

Надається досвід створення репозиторію у системі контролю версій.

Також лабораторна робота дає основні навички розробки з використанням IDE ігрового рушія. Дається можливість роботи з іншим типом IDE за вибором студента та по узгодженню з викладачем.

Для підготовки ЛР № 1 може бути використаний будь-який сучасний ПК на основі мікропроцесора AMD64 (Intel® 64) або ARM. Операційна система Windows, Linux або Mac OS (Macintosh Operating System).

Вхідні дані ЛР 1.

Прізвище студента; ім'я студента; шифр навчальної групи; скорочена назва факультету; скорочена назва університету. Порядковий номер у списку, що визначає варіант.

Вхідні дані ЛР 1.

Репозиторій на GitHub з проектом. У окремому файлі вказана вся первинна інформація, що обговорена у вхідних даних. На даному етапі репозиторій не є обов'язковим. Дозволяється тримати проект локально. У проекті реалізовані всі вимоги відповідно до завдання і варіантів. Проект має запускатися на машині студента і викладача. У разі наявності помилок проект не зараховується.

Завдання

Репозиторій у системі контролю версій. Створити проект 2D. Загальні вимоги.

Акаунт на GitHub, на даному етапі за бажанням. Репозиторій на GitHub з проектом. Назва GameProgLab1Group**Num**, де зафарбовано номер групи.

Установка ігрового рушія. Створений проект IDE (2D) на основі рушія, що містить 1 сцену, ігровий персонаж. Можуть бути включені інші елементи. Розроблений і налагоджений скрипт для управління ігровим персонажем. Достатньо продемонструвати рух ліворуч, праворуч, стрибки, коректну фізику, зупинку перед перешкодою. **Проект розташовано у репозиторій на GitHub**, основна мета полягає у дослідженні і підтвердженні володіння обраною IDE (2D) і технологією розподіленої системи контролю версій.

У разі виконання всіх умов і відмінного захисту надається 12,5 балів.

Варіанти завдань

Для виконання завдання у якості об'єкту, що керується через клавіатуру реалізується примітив. Для парного варіанту - коло, для непарного варіанту - квадрат. Зафарблення квадрату за рішенням студенту.

Далі у якості площини і перешкод обирається елемент з набору асертів відповідно до варіантів, що надані далі. Для отримання доступу до асертів потрібна реєстрація <https://assetstore.unity.com>.

Варіанти лабораторної роботи

1. <https://assetstore.unity.com/packages/2d/characters/simple-2d-platformer-assets-pack-188518>
2. <https://assetstore.unity.com/packages/2d/environments/pixel-platformer-art-pack-165498>
3. <https://assetstore.unity.com/packages/2d/textures-materials/tiles/pixel-tiles-for-side-scroller-141838>
4. <https://assetstore.unity.com/packages/2d/environments/minifantasy-forgotten-plains-super-low-res-2d-pixel-art-by-krish-208907>
5. <https://assetstore.unity.com/packages/2d/free-2d-platform-tileset-206082>
6. <https://assetstore.unity.com/packages/2d/environments/megabot-areas-pack-1-195249>
7. <https://assetstore.unity.com/packages/2d/environments/2d-ice-world-106818>
8. <https://assetstore.unity.com/packages/2d/environments/2d-stone-age-world-platformer-starter-kit-86769>
9. <https://assetstore.unity.com/packages/2d/free-2d-mega-pack-177430>
10. <https://assetstore.unity.com/packages/2d/characters/pixel-adventure-1-155360>
11. <https://assetstore.unity.com/packages/2d/environments/free-platform-game-assets-85838>
12. <https://assetstore.unity.com/packages/2d/characters/pixel-adventure-2-155418>

1.2 Теоретичні відомості

Геймплей (*Gameplay*) – це те, що відрізняє комп'ютерну гру від таких неінтерактивних видів розваг, як книги або кіно. Дивлячись кіно або читаючи книгу, людина є лише стороннім спостерігачем. У випадку з геймплеєм, з'являється можливість взаємодіяти з грою, вирішуючи поставлені геймдизайнером завдання за допомогою ігрової механіки.

Ігрові механіки (*Game Mechanics*) – це добірка правил та петель зворотного зв'язку, які призначені створювати геймплей, що приносить задоволення. Вони є будівельними блоками, які застосовуються і комбінуються з ігровим і не ігровим контекстом (наприклад, квести, одиниці досвіду, досягнення).

Локіт (*Lockit*) – документ, що надається замовником, в якому наведені всі тексти додатка усіма мовами, що підтримуються.

Механіка Point-and-Tap – спосіб керування об'єктами в грі, який реалізується шляхом вибору об'єкта або області, до якого необхідно застосувати (перемістити) цей об'єкт.

Тайлова графіка (*від англ. Tile – плитка*) – метод створення великих зображень (як правило, рівнів в комп'ютерних іграх). Зображення складається з маленьких фрагментів однакових габаритів (патернів), як картина з кахлю – звідси назва.

Текстура (*Texture*) – зображення, що накладається на поверхню моделі для додання їй кольору, забарвлення або ілюзії рельєфу. Приблизне використання текстур можна легко уявити як малюнок на поверхні скульптурного зображення.

Уповільнення (*Slow Down*) – дефект, в основі якого найчастіше лежить пряма прив'язка фреймрейта до анімацій і внутрішньоігрових механік. У разі поганої продуктивності, низький фреймрейт призводить до уповільнення ігрового процесу.

Хінт (*Hint*) – ігрова підказка, що пояснює гравцеві деталі геймплея.

(Див. також підказка)

Хітбокси (*Hitboxes*) – колізії, іноді окремі ігрові об'єкти, які накладаються поверх моделей для реалізації механіки отримання шкоди.

Frame rate – кількість змінюваних кадрів за одиницю часу, стандартна одиниця вимірювання – FPS (Frames Per Second/Кадри за секунду).

(Див. також фреймрейт)

ЛАБОРАТОРНА РОБОТА № 2

Дослідження базового патерну ігрового рушія Unity на прикладі тривимірного ігрового застосунку

2.1 Завдання

Мета роботи: полягає у набутті знань, умінь та навичок з технології розроблення основ проекту з використанням обраної мови програмування у обраній парадигмі. Надається досвід створення репозиторію у системі контролю версій і знання елементів середовища розробки і основи вихідного коду для управління грою.

Також лабораторна робота дає основні навички розробки з використанням IDE ігрового рушія.

Вхідні дані ЛР 2.

Прізвище студента; ім'я студента; шифр навчальної групи; скорочена назва факультету; скорочена назва університету. Порядковий номер у списку, що визначає варіант.

Вхідні дані ЛР 1.

Репозиторій на GitHub з проектом. У окремому файлі вказана вся первинна інформація, що обговорена у вхідних даних. На даному етапі репозиторій не є обов'язковим. Дозволяється тримати проект локально. У проекті реалізовані всі вимоги відповідно до завдання і варіантів. Проект має запускатися на машині студента і викладача. У разі наявності помилок проект не зараховується.

Завдання

Створений проект IDE (3D) на основі рушія, що містить 2 сцени, ігровий персонаж. Можуть бути включені інші елементи. Розроблений і налагоджений скрипт для управління ігровим персонажем. Додані умови переходу між сценами. В якості ігрового персонажа використовується звичайний примітив. Достатньо продемонструвати рух ліворуч, праворуч, стрибки, коректну фізику, зупинку перед перешкодою, набір балів і перехід з одного рівня на інший. Можливо, за бажанням реалізувати інтерфейс гравця.

Проект розташований у репозиторій на GitHub, основна мета полягає у дослідженні і підтвердженні володіння обраною IDE (3D) і технологією розподіленої системи контролю версій. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 25 балів.

Варіанти завдань

Для виконання завдання у якості об'єкту, що керується через клавіатуру реалізується примітив. Для парного варіанту - шар, для непарного варіанту - куб. Забарвлення квадрату за рішенням студенту.

Далі у якості площини і перешкод обирається елемент з набору асертів відповідно до

варіантів, що надані далі. Для отримання доступу до асертів потрібна реєстрація <https://assetstore.unity.com>.

Варіанти лабораторної роботи

1. <https://assetstore.unity.com/packages/3d/props/exterior/pbr-fences-181524>
2. <https://assetstore.unity.com/packages/3d/props/basic-bedroom-starterpack-215986>
3. <https://assetstore.unity.com/packages/3d/vegetation/trees/2022-pbr-xfrogplants-sampler-229007>
4. <https://assetstore.unity.com/packages/3d/environments/unity-terrain-hdrp-demo-scene-213198>
5. <https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-simple-nature-pack-162153>

2.2 Теоретичні відомості

Створена концепція гри, загальний опис, основи геймплею, плани монетизації, ескізний проєкт, обраний дизайн.

1. Концертування (Concept)

На цьому першому етапі команда вигадує концепцію гри, і проводить початкове опрацювання ігрового дизайну. Головна мета даного етапу – це геймдизайнерська документація, що включає Vision (розгорнутий документ, що описує гру, як кінцевий бізнес-продукт) і Concept Document (початкове опрацювання всіх аспектів гри).

Виділення і формальний опис Core Gameplay.

Ігровий цикл (англ. *gameplay loop*, також англ. *core gameplay loop*) - принцип, згідно з яким геймдизайнери задають головний елемент ігрової механіки, який визначає фундаментальний досвід гравця. Один ігровий цикл є дією гравця, результат цієї дії в ігровому світі, реакцію гравця на результат і запит гри на повторення нової дії

це безпосередньо футбольний матч: біг, пас, навіс, простріл, гол. Це футбольні правила, начебто положення поза грою, розмірами поля та кількістю ігрових персонажів на полі.

Виділення і формальний опис Meta Gameplay. Створення і формальний опис моделі монетизації. є поняття, що описує активності поза грою, але що впливають її ігровий процес. Немає єдиного визначення метаігри. Під цим поняттям може матися на увазі аналіз гравцями того, що відбувається в окремих ігрових партіях з метою покращити свої результати або вирішити проблеми в наступних. Крім цього, під метагрою можна розуміти запровадження розробниками додаткових елементів (досягнень[2], доповнень та інших), які є частиною центрального ігрового процесу.

У продуктивній документації геймдизайнер формулює та зберігає свої ідеї. Виконавцю документація дозволяє правильно розуміти свої завдання щодо реалізації продукту. Тестувальник чітко бачить, що і як тестувати. Для Продюсера/ПМа ця документація надає матеріал для формування планів та контролю виконання завдань. А інвестор (особливо на ранніх етапах) отримує розуміння, на що саме він виділяє кошти.

Серед ключових принципів формування продуктової документації слід зазначити: структурованість, захищеність від різночитань, повний опис продукту, регулярну актуалізацію.

2. Прототипування (Prototyping)

Важливий етап проектування будь-якої гри – створення прототипу. Те, що добре виглядає «на папері», не обов'язково буде цікаво в реальності. Прототип реалізується з метою оцінки основного ігрового процесу, перевірки різних гіпотез, проведення тестів ігрових механік, перевірки ключових технічних моментів.

Дуже важливо на етапі створення прототипу реалізовувати лише те, що потрібно перевірити й у стислий термін. Прототип може бути простим у реалізації, т.к. після досягнення поставлених перед ним цілей, він має бути «викинутий». Серйозна помилка розробників-початківців – нести тимчасову інфраструктуру і «милиці» реалізації коду в основний проект.

3. Вертикальний зріз (Vertical Slice)

Мета Вертикального зрізу – отримати мінімально можливу повноцінну версію гри, що включає повністю реалізований основний ігровий процес. У цьому високу якість опрацювання обов'язково необхідно реалізувати лише тих ігрових елементів, які значно впливають сприйняття продукту. При цьому всі базові фічі гри присутні як мінімум у чорновій якості. Реалізовано мінімальний, але достатній реалізації повноцінного ігрового процесу набір контенту (один рівень чи одна локація).

4. Виробництво контенту (Content production)

На цьому етапі виготовляється достатня кількість контенту для першого запуску на зовнішню аудиторію. Реалізуються всі фічі, заплановані до закритого бета-тестування. Це найбільш тривалий етап, який може займати для великих клієнтських проектів рік і більше.

На цьому етапі задіяна найбільша кількість фахівців, які займаються виробництвом всього основного наповнення гри. Художники створюють усі графічні ресурси, геймдизайнери налаштовують баланс та заповнюють конфіги, програмісти реалізують та полірують усі фічі.

5. Friends & Family / CBT (закрите бета-тестування)

На етапі CBT продукт вперше демонструється досить широкому загалу, хоча і лояльному продукту або компанії. Серед найважливіших завдань на цьому етапі виступають: пошук та виправлення гейм-дизайнерських помилок, проблем ігрової логіки та усунення критичних багів. На цьому етапі в грі присутні вже всі ключові фічі, створено достатньо контенту для повноцінної гри тривалий час, налаштовано збір та аналіз статистики. Тестування йде тест-планом, проводяться стрес-тести вже із залученням реальних гравців.

6. Soft Launch/OBT (відкритий бета-тест)

На цьому етапі продовжується тестування гри, але вже на широкій аудиторії. Йде оптимізація під великі навантаження. Гра має бути готова до прийому великого трафіку. У грі реалізований білінг та приймаються платежі.

7. Release

Ключова мета – отримання прибутку. Базовий застосовуваний для оцінки прибутковості критерій: кількість грошей, принесених у середньому одним гравцем за весь час (LTV aka lifetime value), має перевищувати витрати на залучення цього гравця (CPI aka cost per install).

На цьому етапі має бути повністю налагоджено оперування продукту (технічна підтримка, робота з комп'ютерні), дотримуються маркетингові та фінансові плани, ведуться роботи з покращення фінансових показників, активно відпрацьовуються канали залучення трафіку.

Репозиторії з безкоштовними асертами

<https://assetstore.unity.com/packages/2d/characters/simple-2d-platformer-assets-pack-188518>
<https://assetstore.unity.com/packages/2d/environments/pixel-platformer-art-pack-165498>
<https://assetstore.unity.com/packages/2d/textures-materials/tiles/pixel-tiles-for-side-scroller-141838>
<https://assetstore.unity.com/packages/2d/environments/minifantasy-forgotten-plains-super-low-res-2d-pixel-art-by-krish-208907>
<https://assetstore.unity.com/packages/2d/free-2d-platform-tileset-206082>
<https://assetstore.unity.com/packages/2d/environments/megabot-areas-pack-1-195249>
<https://assetstore.unity.com/packages/2d/environments/2d-ice-world-106818>
<https://assetstore.unity.com/packages/2d/environments/2d-stone-age-world-platformer-starter-kit-86769>
<https://assetstore.unity.com/packages/2d/free-2d-mega-pack-177430>
<https://assetstore.unity.com/packages/2d/characters/pixel-adventure-1-155360>
<https://assetstore.unity.com/packages/2d/environments/free-platform-game-assets-85838>
<https://assetstore.unity.com/packages/2d/characters/pixel-adventure-2-155418>
Див. Слайди лекції №-1-4.
2.3 Програма співбесіди до лабораторної роботи № 2
Наведені у додатку

Політика та контроль

Політика навчальної дисципліни (освітнього компонента)

7.1. Загальні положення

7.1.1 За викладачем залишаються право шляхом співбесіди визначати рівень знань, вмінь та навичок студента, у тому числі студентів, що проходили професійні курси розробки КІ. В співбесіду включаються питання, що стосуються знання синтаксису мови програмування, ігрового рушія, термінології КІ, основ організації розробки КІ та бізнес питання ігрової індустрії. Для визначення вищого рівня знань можуть включатися питання з математичного супроводження ігрового рушія.

7.1.2 Перші дві лабораторні роботи виконують всі студенти з метою вивчення основ ігрового рушія для роботи у двовимірному і трьохвимірному варіанті КІ.

7.1.3 Лабораторні роботи 3-8 є поетапною розробкою гри. Для виконання цих ЛР дозволяється використовувати прототип. При використанні прототипу обов'язково вказується джерело, описується первинний варіант гри і внесені зміни у ході дослідження протягом ЛР 4-8. Для визначення рейтингу у ході співбесіди задаються питання, що згадані у п.1.

7.1.4 При чистому плагіаті бали за лабораторні роботи не зараховуються.

7.1.5 Обов'язковими є лабораторні роботи 1-6, при виконанні цих лабораторних робіт студент може отримати оцінку добре. Для підвищеної оцінки студент захищає додатково ЛР 7, 8 або здає залік.

7.2. Загальний опис лабораторних робіт і порядок оцінювання.

7.2.1 Лабораторна робота 1 (обов'язкова). Дослідження базового патерну ігрового рушія Unity на прикладі двовимірної технології

Установка ігрового рушія. Створений проект IDE (2D) на основі рушія, що містить 1 сцену, ігровий персонаж. Можуть бути включені інші елементи. Розроблений і налагоджений скрипт для управління ігровим персонажем. Достатньо продемонструвати рух ліворуч, праворуч, стрибки, коректну фізику, зупинку перед перешкодою.

Проект розташовано у репозиторій на GitHub, основна мета полягає у дослідженні і підтвердженні володіння обраною IDE (2D) і технологією розподіленої системи контролю версій.

У проекті має бути застосований процедурний і ООП варіант скрипт.

У разі виконання всіх умов і відмінного захисту надається 12,5 балів.

7.2.2 Лабораторна робота 2 (обов'язкова). Дослідження базового патерну ігрового рушія Unity на прикладі тривимірного ігрового застосунку.

Створений проект IDE (3D) на основі рушія, що містить 2 сцени, ігровий персонаж. Можуть бути включені інші елементи. Розроблений і налагоджений скрипт для управління ігровим персонажем. Додані умови переходу між сценами. В якості ігрового персонажа використовується звичайний примітив. Достатньо продемонструвати рух ліворуч, праворуч, стрибки, коректну фізику, зупинку перед перешкодою.

Проект розташовано у репозиторій на GitHub, основна мета полягає у дослідженні і підтвердженні володіння обраною IDE (3D) і технологією розподіленої системи контролю версій. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 25 балів.

Базові вимоги до проекту ігрового застосунку (ЛР 3-6)

Деякі лабораторні роботи можуть бути поєднані у єдине ціле, як частина або складова проекту. У цілому повний курс лабораторних робіт вважається захищеним у тому випадку, коли виконані наступні етапи, результатом яких є прототип КІ:

7.2.3 Лабораторна робота 3 (обов'язкова). Дослідження типових патернів у скриптах ігрового застосунку.

Створений проект IDE (2D або 3 D) на основі рушія, що містить 1 сцену, об'єкт і скрипт управління, що виводить повідомлення, по можливості, скрипт керує об'єктом, **проект розташовано у репозиторій на GitHub**.

У проєкті має бути застосований процедурний і ООП варіант скрипт на базі патерну Стан. Основна мета полягає у дослідженні і підтвердженні володіння обраною IDE і технологією розподіленої системи контролю версій. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 37,5 балів.

7.2.4 Лабораторна робота 4 (обов'язкова). Документування концепції гри.

До ЛР3 додати нову сцену і умови переходу між сценами, графічний інтерфейс користувача гри. **Створити коміт проєкту IDE (2D або 3 D), що містить прототип гри.** Провести дослідження і формалізацію загальної концепції гри у вигляді презентації або іншої документації.

Обов'язковим є опис концепції гри, загальний опис геймплею, плани монетизації, ескізний проєкт, обраний дизайн, декомпозиція прототипу гри на механіки, формальний **опис Core Gameplay, Meta Gameplay, бажано у вигляді презентації.**

Основна мета полягає у дослідженні документування концепції гри.

У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 50,0 балів.

7.2.5 Лабораторна робота 5 (обов'язкова). Базові складові ігрового застосунку.

До ЛР4 додати елементи сцени, оточення, ввести додаткові персонажі у гри, освітлення (за потребою), додати нові рівні. Створити елементи графічного інтерфейсу користувача відповідно до доданих елементів. Додавати нових гравців за допомогою графічного меню користувача. Автоматизувати рейтинг гравців. Додати умови переходу користувача з одного рівня на інший, умови набору рахунку. Додати елементи азарту у гру.

Основна мета полягає у дослідженні процесу захоплення грою і створення відповідних програмних складових. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 62,5 балів.(можна більше)

7.2.6 Лабораторна робота 6 (обов'язкова). Бізнесова складова ігрового застосунку.

Додати елементи монетизації, протестувати. Обов'язковим є **окремий коміт проєкту IDE (2D або 3 D), що містить прототипи.** Протестувати елементи монетизації локально. Бажано реалізувати прототип реальної системи оплати і взаємодії з банком.

Основна мета полягає у дослідженні процесу отримання прибутку від КІ і створення відповідних програмних складових. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 75,0 балів.

7.2.7 Лабораторна робота 7. Мережеві елементи ігрового застосунку.

До ЛР6 додати новий функціонал, що дозволяє виконувати гру у мережі. елементи сцени, оточення, ввести додаткові персонажі у грі, освітлення (за потребою), додати нові рівні. Створити елементи графічного інтерфейсу користувача відповідно до доданих елементів. Додати умови переходу користувача з одного рівня на інший, умови набору рахунку, основна мета полягає у дослідженні процесу створення багаторівневої гри за формалізованою загальною концепцією та її математичний опис.

Додатковим елементом дослідження є виведення прототипу гри до релізу, основна мета полягає у дослідженні процесу розгортання гри і доведення її до гравців. Основна мета полягає у дослідженні програмних елементів функціонуванню гри у мережі. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 87,5 балів.

7.2.8 Лабораторна робота 8. Штучний інтелект у ігрових застосунках.

До ЛР7 додати *прототипи штучного інтелекту, окремий математичний опис складових елементів гри, логіки.*

Основна мета полягає у дослідженні штучного інтелекту у гри. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 100 балів.

Залік. У разі бажання здавати залік може бути надано максимально 25 балів.

Система вимог, які ставляться перед студентом:

- відвідування лекційних та лабораторних занять є обов'язковою складовою вивчення матеріалу;
- *у особливі періоди*, що передбачені законодавством, можливий асинхронний спосіб вивчення матеріалу і здавання лабораторних робіт, який передбачає власний графік здавання лабораторних робіт, що не перевищує довжину курсу;
- на лекції викладач користується власним презентаційним матеріалом; відпрацьовує практичну частину; використовує гугл-диск для викладання матеріалу поточної лекції, додаткових ресурсів, лабораторних робіт та інше; викладач відкриває доступ до певної

директорії гугл-диска для скидання електронних лабораторних звітів та відповідей на МКР;

- на лекції заборонено відволікати викладача від викладання матеріалу, усі питання, уточнення та ін. студенти задають в кінці лекції у відведений для цього час;
- лабораторні роботи захищаються у вигляді співбесіди разом з демонстрацією вихідного коду на власній машині, бали за лабораторну роботу враховуються лише за наявності електронного звіту (вихідного коду);
- модульні контрольні роботи пишуться на лекційних заняттях без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається у файлі до відповідної директорії гугл-диск.

Види контролю та рейтингова система оцінювання результатів навчання (PCO)

- студент отримує позитивну оцінку за результатами роботи у семестрі, якщо має підсумковий рейтинг за семестр не менш ніж 60 балів і виконав умови допуску, був присутній на заліку і не був видалений;
- у особливі періоди, що передбачені законодавством, можлива зміна градації балів з виконання лабораторної роботи і асинхронний спосіб вивчення матеріалу;
 - якщо студент виконав умови допуску PCO і має підсумковий рейтинг за семестр менш ніж 60 балів, або хоче підвищити оцінку, то він виконує залікову контрольну роботу або проходить залікову співбесіду, у день заліку;
 - модульні контрольні роботи пишуться на лекційних заняттях із застосуванням всіх доступних матеріалів;
 - заохочувальні бали виставляються за: активну участь на лекціях; участь у факультетських та інститутських олімпіадах з навчальних дисциплін, участь у конкурсах робіт, підготовка оглядів наукових праць; презентацій по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів на більше 10;
 - штрафні бали виставляються за: невчасну здачу лабораторної роботи, затримка більше ніж 3 тижня. Кількість штрафних балів на більше 10.

Система рейтингових балів та критерії оцінювання

Для позитивної оцінки знання синтаксису мов програмування, що використовуються для написання скриптів ІЗ.

Модульні контрольні роботи (Дозволяється об'єднувати дві модульні контрольні роботи у одну, оцінка у такому випадку збільшується у двічі):

- «**відмінно**» – 100% балів, повна відповідь (не менше 90% потрібної інформації) вміння користуватися знаннями для практичного програмування на рівні навичок, вміння вирішити

завдання у декілька варіантів, визначити перевагу і недоліки різних варіантів рішення, вміння читати і використовувати вихідний код, розробляти власні шаблони і нестандартні рішення;

- «**дуже добре**» – 95 % бала, достатньо повна відповідь (не менше 75% потрібної інформації) вміння користуватися знаннями для практичного програмування в цілому на рівні навичок, вміння вирішити завдання, вміння читати і використовувати вихідний код, розробляти власні шаблони і нестандартні рішення;

- «**добре**» – 85 % бала, відрізняти процедурне і ООП рішення, неповна відповідь (не менше 60% потрібної інформації) та незначні помилки, знання технології практичного програмування, вміння читати і використовувати вихідний код для рішення завдання, розробляти і знати технологічний цикл;

- «**задовільно**» – 75 % балів, відрізняти процедурне і ООП рішення, неповна відповідь (не менше 50% потрібної інформації), помилки, що вимагають допомоги викладача, наявність слабких практичних навичок програмування, вміння вирішувати задачу з використанням шаблонного відхідного коду і додаткової літератури (ресурсів), розробляти і знати технологічний цикл прикладних рішень.

- «**незадовільно**» – 0 балів, відповідь не відповідає вимогам до «задовільно».

Лабораторні роботи (Для позитивної оцінки знання синтаксису мов програмування, що використовуються для написання скриптів ІЗ)

- «**відмінно**» – 100% балів, повна відповідь (не менше 90% потрібної інформації) вміння користуватися знаннями для практичного програмування на рівні навичок, вміння вирішити завдання у декілька варіантів, визначити перевагу і недоліки різних варіантів рішення, вміння читати і використовувати вихідний код, розробляти власні шаблони і нестандартні рішення;

- «**дуже добре**» – 95% бала, достатньо повна відповідь (не менше 75% потрібної інформації) вміння користуватися знаннями для практичного програмування в цілому на рівні навичок, вміння вирішити завдання, вміння читати і використовувати вихідний код, розробляти власні шаблони і нестандартні рішення;

- «**добре**» – 85% бала, відрізняти процедурне і ООП рішення, неповна відповідь (не менше 60% потрібної інформації) та незначні помилки, знання технології практичного програмування, вміння читати і використовувати вихідний код для рішення завдання, розробляти і знати технологічний цикл;

- «**задовільно**» – 75% бал, відрізняти процедурне і ООП рішення, неповна відповідь (не менше 50% потрібної інформації), помилки, що вимагають допомоги викладача, наявність

слабких практичних навичок програмування, вміння вирішувати задачу з використанням шаблонного відхідного коду і додаткової літератури (ресурсів), розробляти і знати технологічний цикл прикладних рішень.

- «**незадовільно**» – 0 балів, відповідь не відповідає вимогам до «задовільно».

За кожне заняття запізнення з поданням лабораторної роботи до захисту від встановленого терміну оцінка знижується на 1 бал.

У особливі періоди, що передбачені законодавством, можлива зміна градації балів з виконання лабораторної роботи і асинхронний спосіб вивчення матеріалу.

Викладач, що веде лабораторні роботи може внести свою корекцію у оцінку ЛР.

Додаток

Загальна програма співбесіди до захисту лабораторних робіт

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. Архітектура і приклад 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.