**Одеський національний політехнічний університет**

**Інститут комп'ютерних систем**

**Кафедра інформаційних систем**

**КУРСОВА РОБОТА**

з дисципліни “Об’єктно-орієнтоване програмування”

Тема “Розробка десктопної гри”

Студента (ки) 2 курсу групи

спеціальності 126 – «Інформаційні системи»

Дейнега Д.О.

(прізвище та ініціали)

Керівник Рудніченко М.Д.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали

м. Одеса – 2024 рік

**ЗАВДАННЯ**

НА КУРСОВУ РОБОТУ

(прізвище, ім’я, по батькові) (група)

1. Тема роботи “ Розробка десктопної гри ”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Термін здачі студентом закінченої роботи

3. Початкові дані до проекту (роботи): Початкові дані генеруються системою автоматично. Вони включають базу даних з профілем користувача, його інтересами, та екземпляром рекомендованого матеріалом з інтернету.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які належить розробити): вступ, аналіз специфіки предметної області, порівняння з альтернативами, проектування, опис процесу роботи, висновки, перелік посилань, код програми

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­­\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Завдання видано \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Завдання прийнято до виконання \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЗМІСТ

[ВСТУП 4](#_Toc56777535)

[1 АНАЛІЗ СПЕЦИФІКИ ПРЕДМЕТНОЇ ОБЛАСТІ 5](#_Toc56777536)

[1.1 Особливості розробки десктопної системи рекомендації матеріалу 5](#_Toc56777537)

[1.2 Аналіз аналогів на ринку 6](#_Toc56777538)

[1.3 Використані засоби розробки 11](#_Toc56777539)

[2 ПРОЕКТУВАННЯ ГРИ 14](#_Toc56777540)

[2.1 Написання user stories та формування backlog`у продукту 14](#_Toc56777541)

[2.2 Backlog продукту 16](#_Toc56777542)

[2.3 Розробка прототипів графічного інтерфейсу 16](#_Toc56777543)

[2.4 Виявлення класів і зв’язків за допомогою CRC-карток 22](#_Toc56777544)

[2.5 База даних 24](#_Toc56777545)

[2.6 UML – діаграми 31](#_Toc56777546)

[2.6.1 Діаграма використання 31](#_Toc56777547)

[2.6.2 Діаграма класів 33](#_Toc56777548)

[2.6.3 Діаграма активності 35](#_Toc56777549)

[2.6.4 Діаграма послідовностей 37](#_Toc56777550)

[3 РОЗРОБКА ФУНКЦІОНАЛУ ПРОГРАМИ 42](#_Toc56777551)

[3.1 Розробка рекомендаційної системи 42](#_Toc56777552)

[3.2 Розробка бази даних 44](#_Toc56777553)

[3.3 Демонстрація роботи програми 45](#_Toc56777554)

[ВИСНОВКИ 51](#_Toc56777555)

[ПЕРЕЛІК ПОСИЛАНЬ 53](#_Toc56777556)

**АНАЛІЗ СПЕЦИФІКИ ПРЕДМЕТНОЇ ОБЛАСТІ**

ВСТУП

1. Колекційні карткові ігри (CCI) - один з найпопулярніших жанрів у світі цифрових розваг. Ці ігри поєднують в собі глибоку стратегію, багатий контент і високий рівень взаємодії між гравцями. Вони залучають мільйони гравців по всьому світу і пропонують безліч можливостей для створення унікальних колод, змагань і соціальної взаємодії.

CCIs стала ще більш популярною, ніж раніше. Яскравими прикладами є такі ігри, як Inscryption, Magic: The Gathering та Hearthstone, які пропонують гравцям захоплюючий ігровий процес. Ці ігри демонструють, як працюють карти та різні підходи до створення світів, дозволяючи гравцям знаходити ігри на свій смак.

Наприклад, Inscryption поєднує в собі елементи карткових битв, рольової гри і хоррора, створюючи унікальну атмосферу і захоплюючий ігровий процес. Magic: The Gathering відомий своєю складністю та стратегічною глибиною, що дозволяє гравцям створювати різні варіанти колод та стратегій. Hearthstone, з іншого боку, пропонує більш доступний підхід до карткових ігор, зберігаючи глибину та різноманітність.

Ця курсова робота присвячена розробці вашої власної колекційної карткової гри. У даній статті розглядаються особливості розробки подібних ігор, аналізуються існуючі на ринку рішення і докладно описується використання інструментів розробки, зокрема Unity, C#, Visual Studio Code, Visual Studio Studio Studio і GitHub.

Мета цієї роботи-створити захоплюючу карткову гру, яка пропонує гравцям нові механізми, глибокі сюжети і багатий ігровий досвід. Використовуючи новітні інструменти і технології, ви можете забезпечити високу продуктивність, зручність і безпеку розроблених рішень.

Розробка колекційних карткових ігор вимагає ретельного планування та аналізу, щоб створити цікавий та захоплюючий продукт для гравця. Ця курсова робота призначена для того, щоб показати вам, як ефективно використовувати сучасні інструменти та методи для створення успішної гри.

**АНАЛІЗ СПЕЦИФІКИ ПРЕДМЕТНОЇ ОБЛАСТІ**

1.1 Особливості розробки карткової колекційної гри

Розробка колекційних карткових ігор вимагає створення захоплюючих ігрових вражень, які захоплюють гравців стратегічною глибиною, багатим змістом і інтерактивністю. Основними особливостями розробки таких ігор є:

* Розробка ігрової механіки: вам належить розробити унікальну ігрову механіку, яка включає в себе правила гри, типи карт, їх характеристики і взаємодія між ними.
* Розробка інтерфейсу гри: інтерфейс гри повинен бути інтуїтивно зрозумілим і зручним для користувача і підтримувати всі функції, необхідні для управління колодою і проведення ігрових сесій.
* Графічний дизайн: візуальна складова гри повинна бути привабливою і відповідати загальній тематиці гри. Це включає створення ілюстрацій для карт, стадіонів, анімації тощо.
* Ігровий баланс: важливо забезпечити баланс між різними картами та стратегіями, щоб гра була чесною та цікавою для всіх гравців.

1.2 Аналіз аналогів на ринку

На ринку представлено кілька популярних колекційних карткових ігор, кожна з яких має свої особливості і служить прикладом для розробки власної гри. Серед них варто виділити такі ігри, як Inscryption, Hearthstone, Magic: The Gathering Arena і Legends of Runeterra.

* **Inscryption**: це інноваційна карткова гра з елементами рогалика і хоррора, створена Daniel Mullins Games. У цій грі колекційна карткова Механіка поєднується з атмосферними пригодами і головоломками. Гравці збирають колоду карт, що представляють різних істот і заклинання, і використовують їх для боротьби з супротивниками в темному світі гри. Inscryption відрізняється своїм унікальним стилем і глибоким зануренням в сюжет.
* **Hearthstone**: популярна онлайн-гра від Blizzard Entertainment, що відноситься до жанру цифрових колекційних карткових ігор. Гравці збирають колоду карт, що представляють героїв, заклинання, істот і зброю, і змагаються один з одним в покрокових боях. Hearthstone відомий своєю простотою в освоєнні, але також і складністю в освоєнні, а також частими оновленнями і подіями, які підтримують інтерес гравців.
* **Magic: The Gathering Arena**: цифрова версія культової настільної карткової гри Magic: The Gathering. У цій грі, розробленій Wizards of the Coast, гравці збирають цифрову колоду з карт з різними здібностями і ефектами, в той час як інші гравці Magic: The Gathering Arena пропонують зручний інтерфейс для онлайн-ігор, зберігаючи при цьому складність і стратегічну глибину оригінальної гри. Він надає інтерфейс, який дозволяє створювати, редагувати та керувати вашими програмами.
* **Legend of Rune Terra**: карткова гра від Riot games, заснована на світі League of Legends. Гравці збирають колоду карт, що представляють чемпіонів, заклинання та істот з різних регіонів світу Rune Terra, і змагаються один з одним у стратегічних битвах. У Legends of Runeterra є гнучка Механіка створення колод і часті оновлення, які роблять гру збалансованою і цікавою.

Аналіз цих аналогів показує, що колекційні карткові ігри користуються великою популярністю завдяки своїй глибокій стратегії, багатому контенту і регулярним оновленням. Ці ігри можуть продемонструвати ефективний підхід до дизайну та механіки та послужити натхненням для розробки власної гри.

1.3 Використані засоби розробки

**Unity**

Потужний ігровий движок, який широко використовується для розробки ігор та різних інтерактивних додатків. Він підтримує мову програмування C# і дозволяє створювати кросплатформні програми. Unity має добре розвинену екосистему плагінів та інструментів, що спрощують процес розробки.

Особливості Unity:

* Підтримка 2D і 3D графіки.
* Можливість інтеграції з різними платформами (Windows, macOS, Linux, iOS, Android і т.д.).
* Підтримка технологій віртуальної і доповненої реальності.
* Широкий набір інструментів для розробки користувальницького інтерфейсу.
* Вбудовані функції для управління фізикою, анімацією та звуком.

Переваги:

* Велика спільнота розробників і багато ресурсів для навчання.
* Підтримка широкого спектру платформ.
* Багатий набір вбудованих інструментів та плагінів.

Недоліки:

* Може бути важким для новачків через велику кількість функцій.
* Високі системні вимоги для розробки.
* Деякі функції доступні тільки у платних версіях.

**C#**

Це мова програмування, яка використовується для написання сценаріїв в Unity. Вона забезпечує високу продуктивність і корисна для розробки як ігрових, так і настільних додатків. C# володіє простим синтаксисом і потужними можливостями для роботи з об'єктно-орієнтованим програмуванням.

Переваги:

* + Висока продуктивність.
  + Зручний синтаксис і зрозуміла структура.
  + Широке застосування в різних областях програмування.
  + Підтримка багатьох бібліотек та фреймворків.

Недоліки:

* + Вимагає певного рівня знань і досвіду для ефективного використання.
  + Не така популярна серед новачків, як деякі інші мови програмування.

**Visual Studio Code**

Безкоштовний редактор коду, який підтримує ряд мов програмування, включаючи C#. Він інтегрується з Unity і надає потужні інструменти для налагодження, автозаповнення коду та управління проектами.

Особливості Visual Studio Code:

* + Простий і швидкий редактор.
  + Підтримка розширень для додавання нових функцій.
  + Вбудовані інструменти для налагодження коду.
  + Інтеграція з системами контролю версій, такими як Git.

Переваги:

* + Безкоштовний і легкий у використанні.
  + Велика кількість розширень для різних мов програмування та фреймворків.
  + Швидке і ефективне автозаповнення коду.

Недоліки:

* + Обмежений набір функцій у порівнянні з повноцінними IDE.
  + Може бути менш продуктивним для великих проектів.

**Visual Studio**

Потужне середовище розробки Microsoft, яке надає повний набір інструментів для розробки програмного забезпечення. Воно підтримує мову C# і інтегрується з Unity, що робить його ідеальним для розробки настільних і ігрових додатків.

Можливості Visual Studio:

- Підтримка широкого спектру мов програмування.

- Потужні засоби налагодження та тестування.

- Інтеграція з системою контролю версій.

- Підтримка розробки для різних платформ.

Переваги:

- Потужний інструментарій для професійної розробки.

- Високий рівень інтеграції з іншими продуктами Microsoft.

- Підтримка великих проектів.

Недоліки:

- Високі системні вимоги.

- Деякі функції доступні тільки у платних версіях.

- Може бути складним для початківців.

**Aseprite**

Популярний інструмент для створення піксельної графіки та анімації. Він широко використовується розробниками інді-ігор для створення 2D-спрайтів, анімацій та художніх робіт.

Особливості Aseprite:

- Інтуїтивно зрозумілий інтерфейс для створення та редагування піксельної графіки.

- Підтримка шарів і режимів анімації для створення складних спрайтів.

- Експорт у різні формати зображень, включаючи GIF для анімації.

- Широкі можливості налаштування палітр кольорів та інструментів малювання.

Переваги:

- Легкий у використанні для створення піксельної графіки.

- Доступний за ціною.

- Потужні інструменти для анімації та редагування.

Недоліки:

- Обмежений функціонал для роботи з іншими типами графіки.

- Відсутність деяких розширених функцій, які є в інших графічних редакторах.

**GitHub**

Платформа для спільної розробки програмного забезпечення, заснована на контролі версій Git. Ви можете зберігати свій код у віддаленому сховищі, відстежувати зміни та співпрацювати з іншими розробниками.

Можливості GitHub:

- Зручний веб-інтерфейс для управління репозиторіями.

- Можливість спільної роботи над проектами.

- Інструменти для контролю версій та відстеження змін.

- Інтеграція з різними середовищами розробки.

Переваги:

- Потужні інструменти для контролю версій і спільної роботи.

- Велика спільнота користувачів і багато ресурсів для навчання.

- Безкоштовний для відкритих проектів.

Недоліки:

- Потрібен час для вивчення основ контролю версій.

- Деякі функції доступні тільки у платних версіях.

- Може бути складним для початківців у налаштуванні.

Ці інструменти відкривають широкі можливості для розробки гри з використанням сучасних технологій і забезпечують необхідний функціонал для створення ігор з різноманітними інтерактивними елементами. Використання їх у поєднанні дозволяє розробникам зосередитися на креативному процесі створення гри, мінімізуючи труднощі технічної реалізації.

**ПРОЕКТУВАННЯ ГРИ**