

Рис. 2. Діаграма фабричних класів

Повторне використання коду – оскільки алгоритми шифрування реалізовані як окремі класи, їх можна легко повторно використовувати в інших проєктах.

Завдяки цій архітектурі розробники можуть легко додавати, видаляти або змінювати алгоритми шифрування без необхідності внесення змін у весь код програми. Це робить систему гнучкою, масштабованою та простою у підтримці, особливо для великих проєктів з багатьма залежностями.

### Список використаних джерел

1. Демашкевич А. В., Антонов Ю. С. Розробка програмного забезпечення для приховування повідомлення в цифрових зображеннях за допомогою методу LSB. Матеріали IV Всеукраїнської науково-практичної конференції «Комп'ютерні технології обробки даних». Вінниця, ДонНУ імені Василя Стуса. 2023. С. 242–245.
2. Антонов Ю. С., Римар П. В., Антонова О. Г. Проблема DoS/DDoS атак навчальних ресурсів здобувачами. *Сучасний захист інформації*. 2019. № 4(40). С. 52–62.
3. Tune N., Perrin J. Architecture Modernization. 2024. 488 с.
4. Design Patterns. URL: <https://refactoring.guru/design-patterns>
5. Wong D. Real-World Cryptography. 2021. 400 с.

УДК 004.1

*Явгусішин Б. А., здобувач 4 курсу спеціальності 122 Комп'ютерні науки, Антонов Ю. С., канд. фіз.-мат. наук, доцент, доцент кафедри інформаційних технологій*

## РЕАЛІЗАЦІЯ КАЗУАЛЬНОЇ ГРИ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ З ВИКОРИСТАННЯМ ПЛАТФОРМИ .NET

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Сучасна ігрова індустрія залучає мільйони гравців з усього світу та отримує великі прибутки. Попри глобальну рецесію та зменшення ігрового ринку через введення обмежень у Китаї, ринок був у передбаченому падінні після двох років зростання через пандемію. Наразі ринок стабілізується, що є гарною можливістю для просування власних ігрових додатків [1–2]. Враховуючи те, що на світовому ринку ігрової індустрії найуспішнішим та найприбутковішим сектором є ігри для мобільних пристроїв, тема роботи є актуальною.

Метою роботи є розробка казуальної гри "Code Frost: Cybernetic Battle" на основі платформи .NET з використанням Unity та мови програмування C#.

Останні дослідження вказують на популярність ігрової індустрії та розробки ігрових додатків. Наприклад, у роботі «Розробка мобільної комп'ютерної гри "Морський бій" під платформу Android за допомогою Java» досліджується процес створення гри з використанням мови програмування Java та висвітлюються проблеми, що можуть виникнути під час розробки гри [3]. Інша робота, «Створення інтелектуальної мережевої гри ШАШКИ», досліджує аспекти розробки мережевих ігор, що включають роботу між web-сервером та desktop-додатками користувачів [4].

Внаслідок дослідження була розроблена казуальна гра «Code Frost: Cybernetic Battle». У процесі розробки гри «Code Frost: Cybernetic Battle» було реалізовано ігрові об'єкти з їх логікою взаємодії, ігрові механіки і можливості та інтерфейс користувача, за допомогою яких гравець може взаємодіяти з грою.

З основних ігрових об'єктів, що були реалізовані, варто виділити:

- ігровий персонаж, яким керує користувач;
- різні види ворогів, що з'являються навколо гравця;
- зброя, якою атакують як вороги, так і ігровий персонаж.

Кожен з ігрових об'єктів має власну логіку взаємодії з іншими. Наприклад, гравець може замінити зброю, яку використовує його персонаж, підібравши інший тип зброї з ігрової площини. Зброя знаходиться в рандомізованих місцях цієї площини.

Щодо реалізованих ігрових механік та можливостей варто зазначити такі: переміщення ігрового персонажа за допомогою екранного стіка, що з'являється при дотику до нижньої половини екрана; автоматична атака персонажа по ворогах наявною зброєю; зона хабу, де можна витратити внутрішньоігрову валюту для покращення показників зброї; зменшення здоров'я ігрового персонажа з плином часу для обмеження часу гри в одній сесії; нескінченна ігрова площина, по якій може переміщуватися гравець.

Також важливою частиною реалізації гри є використання патернів проектування. Головною перевагою є забезпечення легкої підтримки та супроводу створеного програмного продукту. Також їх використання пришвидшує тестування та виявлення багів. Основними використаними патернами були:

➤ dependency injection – у програмі було використано фреймворк Zenject, який реалізовує цей патерн;

➤ state – було реалізовано state machine, що контролює основний процес гри. Наприклад, запускає появу ворогів на ігровому полі;

➤ Model-View-Controller (MVC) – використовується у представленні інтерфейсів та вікон користувачу;

➤ Observer – поведінковий патерн, який дає можливість об'єктам стежити й реагувати на події, які відбуваються в інших об'єктах. Реалізовано в деяких класах для кращої незв'язності коду;

➤ Object pool – використовується для появи нових ворогів на ігровій площині.

Було розглянуто основні реалізовані ігрові об'єкти, ігрові механіки і можливості та основні використані патерни під час реалізації казуальної гри «Code Frost: Cybernetic Battle» на основі платформи .NET з використанням Unity та мови програмування C#, та реалізовано програму (рис. 1).

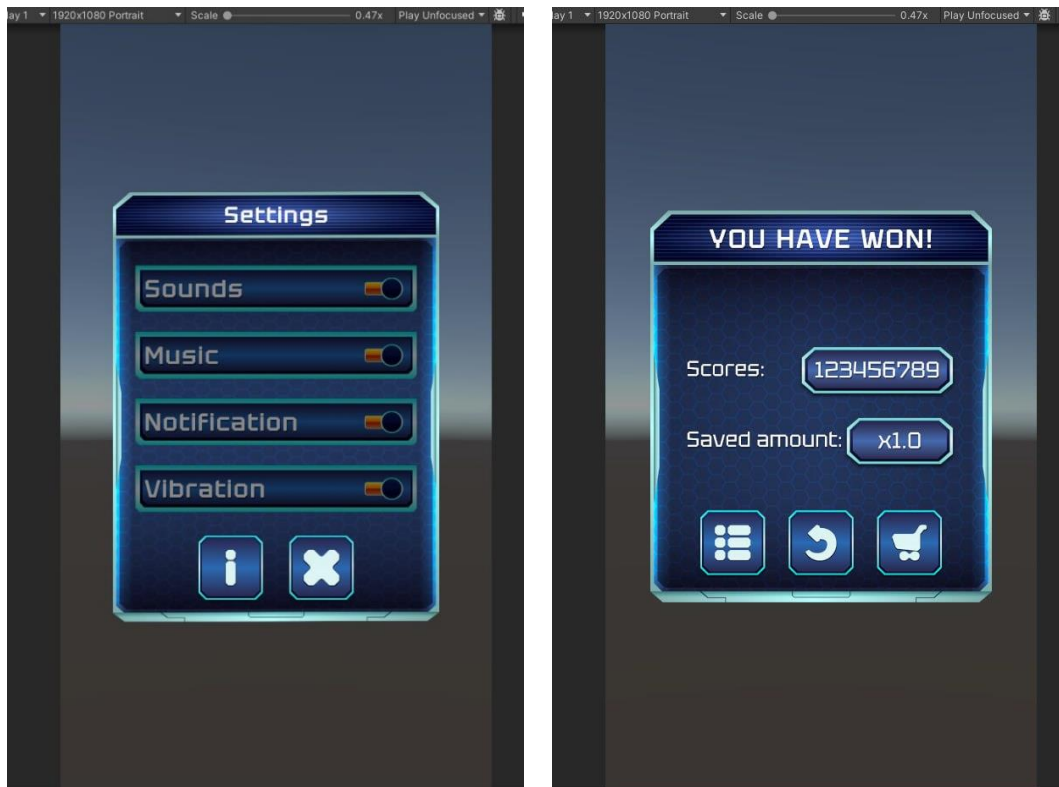


Рис. 1. Приклад інтерфейсу програми

### Список використаних джерел

1. Knezovic A. 200+ Mobile Games Statistics: Market & Revenue Report. 2024. URL: <https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-gaming-statistics> (дата звернення: 15.05.2024).
2. Hawkins A. Chinese gaming sector in turmoil as regulators announce new proposals. 2023. URL: <https://www.theguardian.com/world/2023/dec/27/chinese-gaming-shares-fall-as-regulators-announce-new-proposals> (дата звернення: 15.05.2024).
3. Гнатюк М. А., Антонов Ю. С. Розробка мобільної комп'ютерної гри «Морський бій» під платформу Android за допомогою Java. 2024. URL: <https://jpvs.donnu.edu.ua/article/view/7242> (дата звернення: 17.05.2024).
4. Сірков О. О., Антонов Ю. С. Створення інтелектуальної мережевої гри ШАШКИ. 2024. URL: <https://jait.donnu.edu.ua/article/view/8942> (дата звернення: 17.05.2024).
5. ModestTree. Zenject: вебсайт. <https://github.com/modesttree/Zenject> (дата звернення: 16.05.2024).