

ПРОЄКТУВАННЯ МОДУЛЬНОЇ КРОСПЛАТФОРМНОЇ ГРИ «ШАШКИ» З ВИКОРИСТАННЯМ C# ТА UNITY

Вінницький національний технічний університет

Анотація

У статті описано процес проєктування модульної кросплатформної гри «Шашки» із інтелектуальним ігровим агентом на основі інструментів C# та Unity. Визначено модульну структуру застосунку, що охоплює компоненти управління дошкою, обробки правил, переміщення фігур, обліку результатів та поведінки бота. Ігровий агент визначає припустимі ходи, контролює обов'язкове взяття фігур суперника та автоматично обирає найкращий варіант дії. Отримана система служить основою для подальшого вдосконалення алгоритмів прийняття рішень у логічних настільних іграх.

Ключові слова: C#, Unity, ігровий агент, шашки, модульна архітектура, кросплатформна розробка.

Abstract

This paper presents the design of a modular cross-platform checkers game with an intelligent game agent built using C# and Unity. The modular architecture of the application is defined, covering components for board management, rule processing, piece movement, score tracking, and bot behavior. The game agent evaluates available moves, enforces mandatory captures, and automatically selects the optimal action. The resulting system provides a foundation for future development of decision-making algorithms in logic board games.

Keywords: C#, Unity, game agent, checkers, modular architecture, cross-platform development.

Вступ

Настільні логічні ігри займають особливе місце серед досліджуваних задач сучасного програмування, оскільки поєднують чіткі формальні правила, обмежений простір станів і потребу вироблення оптимальної стратегії. Гра «Шашки» є зручним полігоном для дослідження інтелектуальної поведінки автоматизованого суперника та побудови систем прийняття рішень.

Для реалізації цього продукту обрано платформу Unity з мовою C#, що забезпечує розробку кросплатформних застосунків в межах єдиного проєкту без дублювання коду для кожної платформи. Ця характеристика робить Unity зручним середовищем для проєктування модульних ігрових застосунків.

Основна частина

Програмний застосунок для гри «Шашки» (рис. 1) розроблено на платформі Unity з використанням C#. Unity належить до найбільш популярних ігрових рушіїв і дає змогу виводити один проєкт на кількох платформах — Windows, Android, iOS, Web. Це середовище забезпечує графічний інтерфейс, систему сцен та взаємодію між ігровими компонентами.

Застосунок побудовано за модульним принципом, що дає змогу розділити функціональність на незалежні логічні компоненти. Кожен компонент реалізовано у вигляді C#-скрипта в папці Scripts проєкту Unity.

До основних модулів системи належать:

1. CheckerBoard – відповідає за створення та відображення ігрового поля;
2. RulesManager – забезпечує перевірку правил гри, правильність ходів та можливість взяття шашок;
3. MovementManager – реалізує механізм переміщення ігрових фігур по дошці;
4. GameManager – керує станами гри, початком та завершенням партії;
5. PlayerControl – забезпечує взаємодію гравця з ігровим інтерфейсом;
6. BotControl – реалізує логіку поведінки комп'ютерного суперника;
7. ScoreManager – відповідає за підрахунок результатів гри.

Модуль RulesManager відіграє ключову роль у системі, оскільки він контролює дотримання правил гри. У цьому модулі реалізовано перевірку допустимості ходу, визначення можливості взяття фігури суперника, а також перевірку умов завершення гри.



Рис.1. Головне меню розробленої гри «Шашки»

Особливу увагу в роботі приділено реалізації ігрового агента, який виступає комп'ютерним суперником для користувача (рис.2).

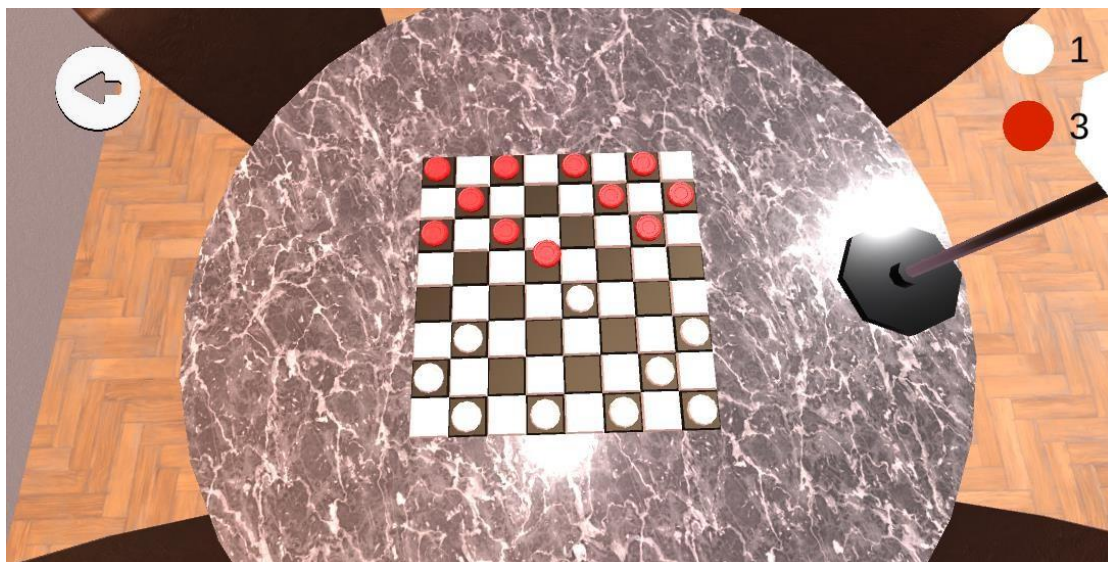


Рис. 2. Процес гри з використанням ігрового агента

Логіка агента реалізована у скрипті BotControl. Алгоритм роботи агента включає декілька етапів:

1. Аналіз поточного стану ігрового поля.
2. Визначення всіх можливих ходів для доступних фігур.
3. Перевірка наявності обов'язкового взяття шашки суперника.

4. Вибір оптимального або випадкового допустимого ходу.
5. Виконання вибраної дії.

Якщо на ігровому полі присутні можливості для взяття фігури суперника, агент обирає саме цей варіант ходу, що відповідає правилам гри. У разі відсутності таких можливостей агент обирає один із допустимих ходів для переміщення фігури.

Реалізований підхід формує базову модель поведінки комп'ютерного суперника, що аналізує стан гри та вибирає дію згідно з регламентом. Перспективою розвитку є впровадження алгоритмів Minimax, Alpha-Beta pruning або методів машинного навчання.

Графічна частина застосунку реалізована з використанням інструментів Unity, що дозволяє відобразити ігрову дошку, фігури та інтерфейс користувача (рис.3).



Рис. 3. Ігрове поле програми «Шашки»

Інтерактивна взаємодія гравця з системою здійснюється через обробку подій натискання миші та взаємодію з об'єктами сцени.

Перевагою використання Unity є підтримка кросплатформної розробки. Це дозволяє компілювати один і той самий проєкт для різних операційних систем без значних змін у коді. Завдяки цьому розроблений програмний продукт може бути адаптований для різних пристроїв.

Розроблена система має модульну структуру, що забезпечує можливість її подальшого розширення. Зокрема, можливе додавання нових режимів гри, реалізація різних рівнів складності комп'ютерного суперника, а також впровадження мережевого режиму гри між кількома користувачами.

Висновки

За результатами роботи спроектовано і реалізовано модульний кросплатформний застосунок гри «Шашки» на основі C# та Unity. Система охоплює візуалізацію ігрового поля, обробку правил, переміщення фігур та взаємодію користувача з інтерфейсом.

Реалізована модульна архітектура охоплює незалежні компоненти: управління дошкою, контроль правил, обробка ходів, облік результатів та логіка бота. Це спрощує подальше розширення функціональності та супровід програмного коду.

Окрему звернено увагу на проєктування ігрового агента як автоматизованого суперника. Алгоритм бота ґрунтується на оцінці стану поля, формуванні можливих ходів і виборі найприйнятнішого варіанта дії, що забезпечує основні ознаки інтелектуальної поведінки.

Практична цінність роботи полягає у створенні повнофункціонального кросплатформного програмного

застосунку, який може бути використаний як основа для подальших досліджень у галузі розробки ігрових систем та штучного інтелекту. У подальшому можливе вдосконалення системи шляхом реалізації більш складних алгоритмів прийняття рішень, додавання різних рівнів складності гри та розширення можливостей програмного застосунку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Unity Technologies. Unity User Manual [Електронний ресурс]. – Режим доступу : <https://docs.unity3d.com/Manual/index.html>
2. Unity Technologies. Unity Scripting API [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/ScriptReference/>
3. Millington I. Artificial Intelligence for Games. – Boca Raton: CRC Press, 2019. – 900 p.
4. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. – 4th ed. – Pearson, 2021. – 1136 p.
5. Aleksandr3377. Checkers Game Project [Електронний ресурс]. – Режим доступу: https://github.com/Aleksandr3377/Checkers?utm_source=chatgpt.com

Козловський Олександр Сергійович - студент групи ІІСТ-22б, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: sk@vin.ua

Богач Ілона Віталіївна — к.т.н., професор кафедри Автоматизації та інтелектуальних інформаційних технологій, Вінницький Національний Технічний Університет, Вінниця, e-mail: ilona.bogach@gmail.com.

Kozlovskyi Oleksandr S. — student, group IIST-22b, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: sk@vin.ua.

Bogach Ilona V. — Ph.D., Professor at the Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com.