

ВИКОРИСТАННЯ UNITY ТА МОВИ C# ДЛЯ РОЗРОБКИ КРОСПЛАТФОРМНОЇ ГРИ «ШАШКИ»

Вінницький національний технічний університет

Анотація

У роботі розглянуто особливості розробки кросплатформної інтелектуальної системи ігрового агента для гри «Шашки» з використанням середовища Unity та мови програмування C#. Запропоновано програмну структуру застосунку, яка включає модулі керування дошкою, ігровими правилами, ходами, підрахунком результатів та логікою бота. Реалізований ігровий агент базується на аналізі допустимих ходів, контролі обов'язкового взяття шашок суперника та автоматизованому виборі дії. Отриманий результат може бути використаний як основа для подальшого вдосконалення алгоритмів прийняття рішень у настільних іграх.

Ключові слова: Unity, C#, шашки, ігровий агент, кросплатформна розробка, штучний інтелект.

Abstract

The paper considers the features of developing a cross-platform intelligent game agent system for the game of checkers using Unity and the C# programming language. The software structure of the application is proposed, including modules for board control, game rules, moves, score management, and bot logic. The implemented game agent is based on the analysis of legal moves, mandatory capture control, and automated action selection. The obtained result can be used as a basis for further improvement of decision-making algorithms in board games.

Keywords: Unity, C#, checkers, game agent, cross-platform development, artificial intelligence.

Вступ

Розробка інтелектуальних ігрових систем є актуальним напрямом сучасної комп'ютерної інженерії та програмування. Особливий інтерес становлять настільні логічні ігри, оскільки вони поєднують чітко формалізовані правила, обмежений простір станів і необхідність вибору раціональної стратегії. Однією з таких ігор є шашки, які є зручним середовищем для дослідження поведінки ігрових агентів та побудови програмних систем прийняття рішень.

Для реалізації такого програмного продукту доцільно використовувати Unity, оскільки цей рушій підтримує C#-скриптинг і дозволяє створювати ігри для різних платформ у межах єдиного проєкту. Це робить його зручним інструментом для створення кросплатформних ігрових застосунків.

Основна частина

У межах виконання роботи було розроблено програмний застосунок для гри «Шашки» (рис. 1) з використанням середовища розробки Unity та мови програмування C#. Unity є одним із найбільш поширених ігрових рушіїв, що забезпечує створення інтерактивних програм для різних платформ, зокрема Windows, Android, iOS та Web. Використання цього середовища дозволяє реалізувати графічний інтерфейс, фізику об'єктів, систему сцен та взаємодію між компонентами гри.

Архітектура розробленого програмного застосунку побудована за модульним принципом, що дозволяє розділити функціональність системи на окремі логічні компоненти. Основні модулі реалізовані у вигляді скриптів на мові C#, які розміщені у папці Scripts Unity-проєкту.

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).

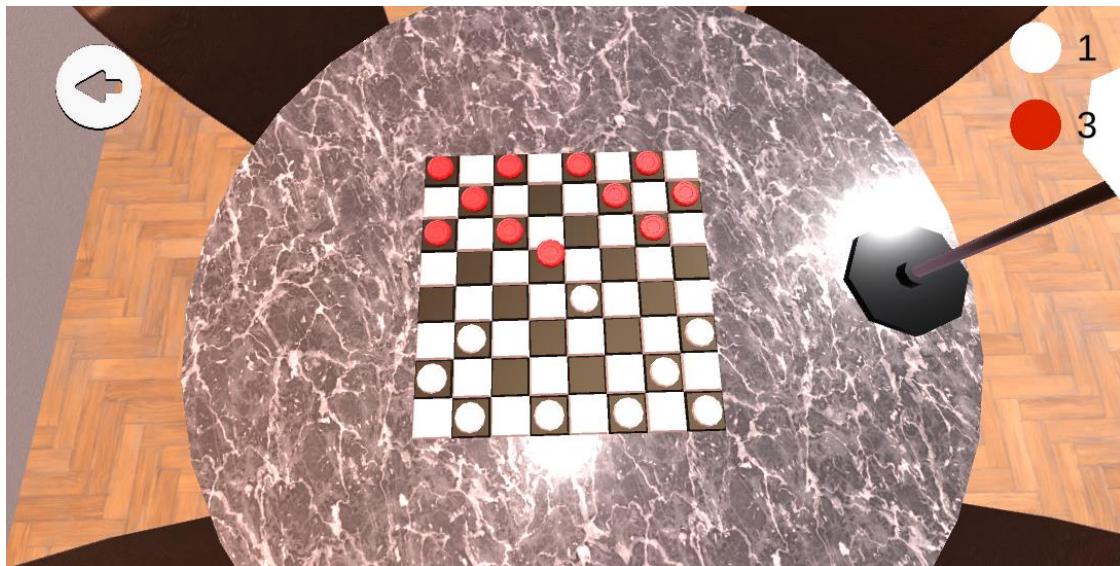


Рис. 2. Процес гри з використанням ігрового агента

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.

Такий підхід дозволяє реалізувати базову модель інтелектуального ігрового агента, який аналізує поточний стан гри та приймає рішення відповідно до правил. У подальшому дану систему можна вдосконалити шляхом впровадження більш складних алгоритмів прийняття рішень, зокрема 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

У результаті виконаної роботи було розроблено програмний застосунок гри «Шашки» з використанням середовища розробки Unity та мови програмування C#. Реалізована система забезпечує відображення ігрового поля, керування ігровими фігурами, перевірку дотримання правил гри та взаємодію користувача з програмним середовищем.

У процесі розробки було створено модульну архітектуру застосунку, що включає компоненти для керування ігровою дошкою, контролю правил, обробки переміщення фігур, управління станом гри та підрахунку результатів. Такий підхід забезпечує зручність подальшого розширення функціональності системи та спрощує підтримку програмного коду.

Особливу увагу приділено реалізації ігрового агента, який виступає комп'ютерним суперником для користувача. Запропонований алгоритм роботи агента базується на аналізі поточного стану ігрового поля, визначенні можливих ходів та виборі допустимого варіанту дії відповідно до правил гри. Це дозволяє реалізувати базову модель інтелектуальної поведінки комп'ютерного суперника.

Практичне значення роботи полягає у створенні функціонального 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

Телега Денис Ігорович – студент групи ІІСТ-22б, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: denistelega09@gmail.com.

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

Kozlovskiy Oleksandr S. — student, group 1IST-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.

Tieliega Denys I. – student of 1IST-22b group, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: denistelega09@gmail.com.

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.