

**Рувінська В.М.**

Національний університет «Одеська політехніка»

**Тройніна А.С.**

Національний університет «Одеська політехніка»

## ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ (МЕТОДИКА) РОЗРОБКИ НАВЧАЛЬНИХ КОМБІНАТОРНИХ ІГОР: ПРОДОВЖЕННЯ

*У наш час розроблені програмні системи лише для найбільш популярних і складних комбінаторних ігор, таких як шахи, шашки, тощо. Запропоновано створювати такого роду системи і для простих неупереджених ігор, зокрема “НІМ”, “НІМ з обмеженнями” і “Дати”, що дозволяють користувачам не тільки грати, а також вивчати теорію комбінаторних ігор. Основною метою роботи є скорочення часу на створення навчальних комбінаторних ігор, для цього створена методика (ІТ) їх розробки, що має чотири етапи, такі як аналіз вимог, проектування, реалізація і тестування, як і для будь-якої програмної системи, але всі вони мають узагальнений характер, тобто підходять для розробки різних комбінаторних ігор. Ця стаття є другою в циклі із двох статей на цю тему, в першій були докладно описані початкові етапи технології, а в цій наведено останні кроки, тобто загальна MDA-структура на етапі аналізу вимог; опис того, як необхідно виконувати кожний етап загальної методики при розробці конкретної навчальної комбінаторної гри; створення конкретних комбінаторних ігор, а також аналіз, чи значно запропонована технологія скоротила час на їх розробку. В результаті було реалізовано програмну систему з використанням створеної методики (ІТ): розроблено три прості комбінаторні гри: “НІМ”, “НІМ з обмеженнями”, “Дати”, що не тільки грають з користувачем, а й вчать грати оптимально, тобто допомагають визначити виграні та програшні стани гри, дають підказки, як правильно ходити, тощо. При реалізації ігор були використані Unity2D, NetBeans IDE для Java, git. Показано, що час створення кожної з ігор завдяки узагальненій методиці зменшився в середньому в два рази у порівнянні зі розробкою кожної окремо.*

**Ключові слова:** методика розробки навчальних комбінаторних ігор, математична теорія комбінаторних ігор, “НІМ”, “НІМ з обмеженнями”, “Дати”.

**Постановка проблеми.** Створення ігор – складний процес, достатньо трудомісткий і потребує залучення висококваліфікованих розробників. Якщо мова йде про створення навчальної гри, то штат розробників необхідно доповнити спеціалістами в області освіти, а також спеціалістами в предметній області. До навчальних ігор повинні застосовуватися такі критерії якості, як і до інших комп’ютерних ігор: захоплюючий геймплей, привабливість ігрового світу, сюжет та стиль гри, високий рівень видимого та звукового оформлення, зручність взаємодії з гравцем. Недооцінка цих якостей при створенні гри може призвести до повної втрати мотиваційного ефекту, що веде за собою зниження ефективності гри як навчального інструменту.

Існують програмні системи, які дозволяють користувачам грати в комбінаторні ігри та вдосконалювати свої навички, але насамперед для найбільш популярних, таких, як, наприклад, як шашки та шахи. Тому актуальним напрямком

є створення систем, що навчають і іншим, зокрема, простим комбінаторним іграм на прикладі ігор “НІМ” і “Дати”. Для цього необхідно запропонувати загальні підходи до створення комбінаторних ігор, а саме, провести аналіз вимог до комбінаторної гри, використовуючи такі документи, як дизайн-документ гри та MDA-структуру. І далі на цій основі провести проектування, реалізацію та тестування запропонованих рішень.

**Аналіз останніх досліджень і публікацій.** *Історія розвитку комбінаторної теорії ігор.* Існують чотири основні посилання на матеріали щодо комбінаторних ігор. Одна з них – це дослідницька книга «On Numbers and Games» [1, с. 3-8], Дж. Конвей, видана в Лондоні в 1976 р. і перевидана потім в 1979 р., а потім і в 2000 р. Ця книга містить багато основних положень за цією тематикою і призвела до швидкого розвитку цього напрямку, який триває й донині. Ще одна книга – двотомне видання «Winning ways for your mathematical plays» [2, с. 14-15], написана

Берлекемпом, Конвеем і Гаєм у 1982 р. і перевидана в 2001 р. У книзі описано багато цікавих ігор, більша частина з яких доступна студентам-бакалаврам математичного спрямування. Найпростіше введення в рівноправні комбінаторні ігри представлено в книзі Richard K. Guy «Fair Game» (у перекладі, «Чесна гра») [3] опублікована в 1989 році. Сучасна книга з комбінаторної теорії ігор «Thomas S. Ferguson. Game Theory» [4, с. I-3 – I-5], видана у 2014 році. З наведених вище прикладів можна зробити висновок, що розвиток теорії комбінаторних ігор не стоїть на місці.

Критичний огляд навчальних комбінаторних комп'ютерних ігор. Існують ігри [5, 6], що навчають користувача грати в шахи з нуля або допомагають покращити наявні навички гри завдяки різним рівням складності. Вони надають користувачеві можливість грати в шахи проти комп'ютера зі зростаючим рівнем складності. Також наявний шаховий наставник, що слідкує за ходами користувача та робить підказки. Є шашкова програма [7], що об'єднує в собі можливості гри і роботи з шашковими базами партій. Комп'ютерна програма Go [8] надає можливість зіграти партію з попередньо заданими параметрами, включає більше 2000 задач, різних по складності. Програма для гри у реверсі [9] ведеться на заданій користувачем складності: від новачка до професіонала.

Описані аналоги застосовані по-перше, лише для популярних ігор; по-друге, користувачам із-за складності кожної з цих ігор не надається можливість вивчати загальні основи теорії комбінаторних ігор; і третє, оскільки ігри, що розглядаються, відносяться до одного класу, то і можна узагальнено проводити їх розробку вже починаючи з етапу визначення вимог до програмної системи.

В роботі описані створені програмні системи для навчання комбінаторній теорії ігор, а також узагальненні способи їх проектування на прикладі простих комбінаторних ігор, таких як “НІМ”, “НІМ з обмеженнями”, “Дати”.

**Постановка завдання.** Метою дослідження є розробка методики зі скорочення часу на побудову комп'ютерних навчальних комбінаторних ігор на базі узагальненої інформаційної технології їх розробки, що запропонована в роботі.

**Виклад основного матеріалу.** В [10] наведено інформаційну технологію для створення різноманітних комбінаторних ігор, а також деталізовано основні, зокрема, початкові етапи. Далі розглянемо докладно інші етапи, що не були детально описані раніше, а саме: дизайн-документ для кон-

кретної комбінаторної гри, узагальнену MDA-структуру для комбінаторної гри, проектування сховища даних, етапи реалізації та тестування (нижченаведена нумерація відповідає номерам методики [10]).

Скорочений дизайн-документ для гри «НІМ з обмеженнями». Створюється на базі узагальненого дизайн-документу для комбінаторних ігор [10]. “НІМ з обмеженнями” – гра, в якій два гравці по черзі беруть предмети із купки. За один хід з купки можна взяти тільки стільки об'єктів, скільки задано в параметрах гри. Програє гравець, який не може зробити хід.

Кожна купка представляється на екрані у вигляді одного рядка предметів. Діями гравця є вибір купки та обрання кількості предметів, які треба з неї взяти (тобто видалити), та передача ходу іншому гравцеві. Його мета – це перемога у грі, досягти якої можна залишивши після свого ходу таку позицію, в якій другий гравець програє. Перешкодою у грі є наявність другого гравця, мета якого збігається з метою першого. Ігрова механіка передбачає взаємодію гравця з ігровими об'єктами, змінюючи стан гри. Система генерує об'єкти на полі (купки) і заповнює їх предметами. Купки заповнюються предметами у відповідності до обраних параметрів. При спробі обрання іншої кількості предметів, що не збігається із заданою в параметрах гри, система повинна попередити користувача та вивести доступні йому ходи.

Набір ігрових предметів та елементів інтерфейсу зображений на рис. 1, в якому присутні кнопки для відкриття вікна, паузи та передачі ходу іншому гравцеві. Предмети, що може обрати користувач, можуть знаходитись у двох станах: активний – об'єкт буде вважатися обраним та змінювати своє положення один раз у декілька кадрів після кліку по ньому; звичайний – об'єкт зберігає статичне положення на полі, оскільки не був обраним гравцем. Для завершення ходу гравцеві необхідно натиснути кнопку передачі ходу. Також йому доступні елементи для переходу між сценами.

Mechanics-Dynamics-Aesthetics (MDA) структура. В ігровому дизайні структура MDA – це інструмент, який використовується для аналізу та формалізує використання ігор, розбиваючи їх на три компоненти [11, с. 1-4]. MDA-структура уточнює терміни «ігровий процес» (gameplay) і «забава» (fun), використовує їх визначення для демонстрації стимулюючих і дестимулюючих властивостей різної динаміки за різними типами естетичного використання гри. З точки зору

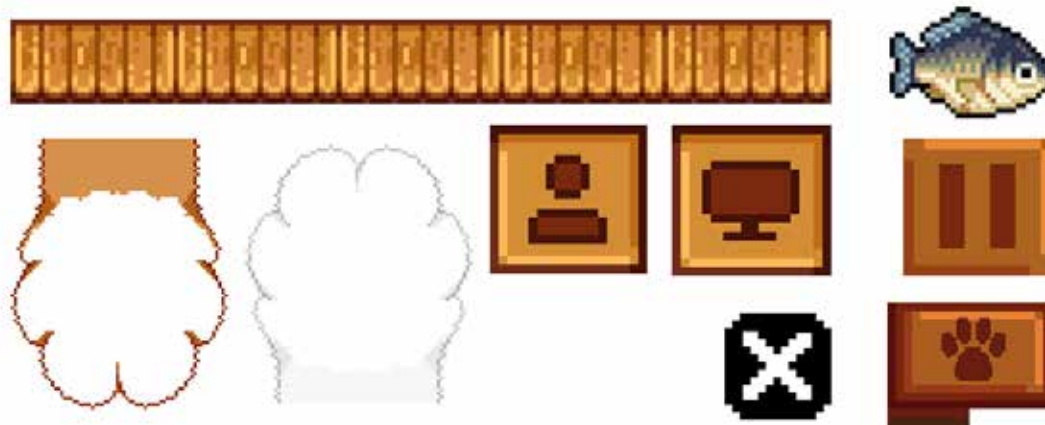


Рис. 1. Набір ігрових об'єктів гри «НІМ з обмеженнями»

дизайнера, механіка створює динаміку, яка, в свою чергу, генерує естетику. Такий напрям зв'язку часто створює проблему для дизайнера ігор, тому що він здатний впливати тільки на механіку, і тільки через неї можна отримати значущу динаміку і естетику для гравця. Але гравець бачить гру через роботу трьох компонент: правила, система, «забава», тому для гравця використовується зворотній напрям зв'язку: він відчуває гру через естетику, що надається ігровою динамікою, яка з'являється з механіки.

Компонент «Естетика» визначає емоційні реакції, «забаву» для гравця. Існують вісім видів естетики: відчуття (Sensation), фентезі (Fantasy), розповідь (Narrative), виклик (Challenge), братерство (Fellowship), дослідження (Discovery), вираз (Expression), підпорядкування (Submission). Навчальні комбінаторні ігри для гравця – це перешкода, що закликає його щось подолати, освоїти, підвищуючи при цьому ймовірність повторного використання гри. Такі ігри мають свій рівень «забави», що формується такими видами естетики як: «відчуття» – гравець, що не грав у такий тип ігор, відчуває щось зовсім незнайоме; «виклик» – гра закликає освоїти математичну теорію ігор, інакше ти не зможеш виграти; «підпорядкування» – гра як проведення часу.

Компонент «Механіка» включає: game-правила, тобто правила, цілі, що визначають мету гри; play-правила, тобто правила маніпуляції, що визначають основні дії, які гравець може застосувати в грі; мета-правила, що визначають способи налаштування або зміни гри, а також алгоритми і структури даних в ігровому рушії [12, с. 2-4]. Для комбінаторних ігор game-правила співпадають з класичними правилами для цих ігор. В якості play-правил можна виділити такі дії, як *множинний вибір об'єктів*, використовуючи будь-який

пристрій введення, тобто комп'ютерну миш або Touchpad та установлення або утримання декілька об'єктів в певному стані; *запис*, тобто, введення символічних значень з клавіатури; *погодження*, коли гравець повинен встановити програшну позицію для суперника. До мета-правил відноситься *рандом*, тобто створення кожного разу нової гри; способи налаштування складності гри та інше.

Компонент «Динаміка» створюється в процесі гри як результат, який є часто непередбаченим в процесі початкової розробки поведінки компонента «Механіка», викликаних: діями гравця, які залежать від його психо-фізіологічних здібностей, а також результатами взаємодії різних механік між собою. Динаміка для комбінаторних ігор досягається за рахунок підвищення складності гри у процесі навчання. Вид Естетики «виклик» створюється такими динамічними характеристиками як вплив часу і гра суперника, яку є бажання перевершити, а також результат взаємодії механік «вибір», «запис» та «погодження». В табл. 1 зведено результати MDA-аналізу.

Проектування сховища даних для збереження результатів проведених ігор, які можуть бути використані під час аналізу успішності навчання. На рис. 2 представлена концептуальна модель даних у випадку, якщо застосовуються БД. Атрибутами сутності для «Користувач» є логін та пароль для входу у власний профіль. Користувач отримує дані з сутності «Результати гри G» у вигляді статистики, де G – розроблювана комбінаторна гра. Атрибутами цієї сутності виступає перелік характеристик з наступних характеристик: результат (перемога чи поразка), складність гри та час партії. Замість використання сервера та бази даних розробник може зберігати дані про проведені ігри у локальному сховищі, наприклад, використати файл XML (рис. 3).

Результати аналізу гри за допомогою MDA-структури

MDA-компонент	Що включає для комбінаторних ігор
Естетика	«виклик» (Challenge) «відчуття» (Sensation) «підпорядкування» (Submission)
Механіка	«вибір» (Select) «запис» (Write) «рандом» (Random) «погодження» (Match)
Динаміка	Вплив часу, гра противника та динаміка, що утворюється у результаті взаємодії механік «вибір», «запис» та «погодження» під впливом психо-фізіологічних здібностей користувача.

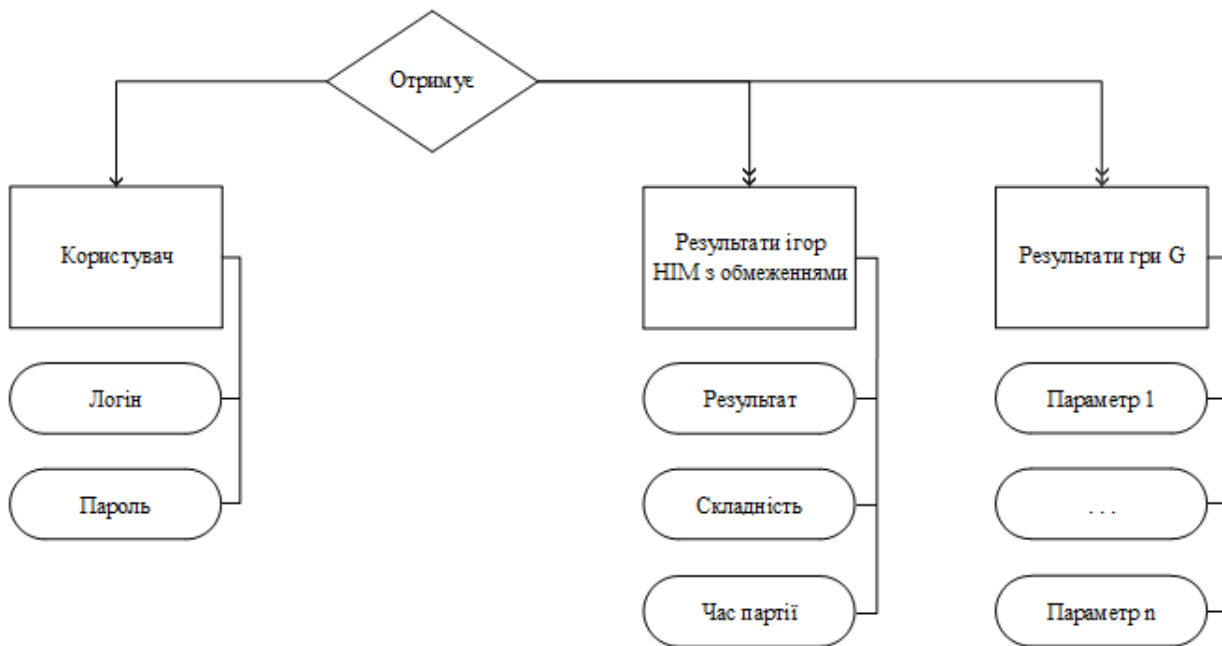


Рис. 2. Концептуальна модель даних

*Проектування конкретних комбінаторних ігор* згідно з узагальненою інформаційною технологією для створення комбінаторних ігор [10]. В результаті повинні бути розроблені необхідні для неї алгоритми (рис. 4).

Користувач обирає параметри гри, а саме: власника першого ходу; кількість купок; значення, що вказують на кількість предметів, яку можна взяти за один хід. Далі система генерує та зберігає позицію гри, а також виводить її гравцеві. Починається цикл гри. За один прохід циклу гравець повинен зробити хід та передати хід іншому гравцеві, а система обчислить змінений стан гри. У разі, якщо гравець залишає після свого ходу кінцеву позицію, то гра завершується, і система виводить інформаційне вікно з переможцем партії.

Далі наводиться детальна структура БД; архітектура, згідно узагальненій діаграмі компонентів

Структура XML-файлу для збереження даних:

```

<results>
  <game>
    <gameName>game_name</gameName>
    <gamePart>
      <parameter1>value1</parameter1>
      <parameter2>value2</parameter2>
      <parameterN>valueN</parameterN>
    </gamePart>
    <gamePart>...</gamePart>
  </game>
</results>

```

Рис. 3. XML для локального сховища



Рис. 4. Алгоритм гри «НІМ з обмеженнями»

тів з додаванням до неї необхідних додаткових класів, полів та методів, властивих цій грі; також детальні макети інтерфейсу користувача.

**Реалізація.** На цьому етапі створюється весь контент для гри. Реалізуються усі функції, що були визначені на етапі аналізу вимог та проектування. Розробка основного наповнення гри передбачає створення графічних ресурсів та реалізацію зазначених алгоритмів гри.

Для створення графіки пропонується використовувати Photoshop або Gimp, для моделювання та анімації Maya або Blender. Для полегшення розробки гри слід використати ігровий движок, що являє собою центральну програмну частину відеогри, яка відповідає за всю її технічну сторону, систематизує її внутрішню структуру. Це може бути Unity3D, Unreal Engine, Construct.

Методика (IT) була апробована в розробці ігор за допомогою Unity3D, оскільки даний ігровий движок надає функціональні можливості, які дозволяють їх використання в різних іграх. В Unity присутня візуальна розробка, а також підтримка багатьох платформ. Також Unity дозволяє використання необхідного розробникові способу контролю версій. Програмна складова реалізується за допомогою скриптів, їх написання вико-

нується з використанням мови програмування C# або JavaScript [13, с. 235-252]. Редагування коду відбувається з використанням зовнішнього інтегрованого середовища розробки Visual Studio або MonoDevelop.

Якщо під час етапу проектування було прийнято рішення про розробку сервера, то він повинен підтримувати багатопотоковість та формувати запити до БД відповідно до запитів клієнта, для цих цілей підійде ряд наступних технологій: MySQL – вільна СУБД для реляційних БД; HeidiSQL – вільний відкритий клієнт для управління БД, підтримує з'єднання та роботу з MySQL; NetBeans IDE – вільне інтегроване середовище розробки для Java, що підтримує рефакторинг, профілювання, виділення синтаксичних конструкцій, автодоповнення мовних конструкцій, шаблони коду та інше.

**Тестування.** Була розроблена система «Game of Strategy», що дозволяє користувачу грати в три комбінаторні гри: «НІМ», «НІМ з обмеженнями» та «Дати». Проведена перевірка розробленої системи на предмет відповідності вимогам, заявленим у узагальненому дизайн-документі та дизайн-документах розроблених ігор, а також в описі MDA-структури.

При вході в систему новий користувач проходить реєстрацію. В подальшому авторизується, вибирає гру та її параметри, а саме, хто робить перший хід, кількість купок, складність, для гри «НІМ з обмеженнями» ще задає, скільки предметів можна взяти із купки за хід.

Якщо вибрані «НІМ» або «НІМ з обмеженнями», то попадає в поле гри (рис. 5). Тут є п'ять купок, кожна з яких представлена рядком із рибок.

Під час гри користувачеві доступні перегляд попередніх ходів (для цього зліва видно, що вже були 4 ходи системи і гравця, і можна натиснути на ці піктограми та їх побачити), вікно паузи (в



Рис. 5. Сцена гри «НІМ» і «НІМ з обмеженнями»

правому верхньому куті), під час якої можна переглянути правила гри, а також для гри «НІМ» та «НІМ з обмеженнями» – блокнот для розрахунків. Для гри «НІМ з обмеженнями», коли користувач помилково намагається взяти кількість предметів, відмінну від встановленої, система видає підказку про доступні ходи.

Якщо вибрана гра «Дати», то вибираються початкова та кінцева дати. У цій грі змінюється лише ігрове поле, інші елементи інтерфейсу однакові для усіх ігор (рис. 6). Видно, що початкова дата – 10 жовтня, а кінцева дата – 16 жовтня. І під час навчання гравцю показують кольором виграшність позицій: жовтий колір – програшна, синій – виграшна. Дійсно, 16 жовтня – це програшна позиція для першого гравця, тому що він не може зробити хід, бо далі не можна ходити, так як це кінцева дата. 15 жовтня – теж програшна позиція для першого гравця: перший гравець почне ходити, і він може збільшити початкову дату тільки на 1 день, таким чином, він одержує 16 жовтня, тобто програш. А 14 жовтня – виграшна позиція для першого гравця, тому що гравець змінить число на 1, і таким чином другий гравець попаде на 15 жовтня, тобто в програшну позицію.



Рис. 6. Сцена гри «Дати»

По завершенню гри користувачеві пропонується повернутися до головного меню або зіграти ще раз з такими ж параметрами. Після повернення до головного меню користувач може змінити параметри, переглянути статистику своїх ігор, а також список найкращих гравців з їх досягненнями.

**Висновки.** Стаття є завершальною із циклу з двох статей, що присвячені узагальненій інформаційній технології створення навчальних комбінаторних ігор. Проведено докладний опис останніх етапів, що не були описані в першій статті, а саме, MDA-структура для формалізації можливостей використання комбінаторних ігор, проектування сховища даних для збереження результатів проведених ігор, рекомендації з реалізації комбінаторних ігор, дизайн-документ та схема алгоритму для гри «НІМ з обмеженнями», тестування. З використанням створеної ІТ (методики) було розроблено програмну систему, що дозволяє користувачеві вчитися теорії та грати в прості комбінаторні ігри «НІМ», «НІМ з обмеженнями» та «Дати», а також задати параметри гри, зберегти результати партій, поглянути статистику по партіям та кращих гравців.

Згідно з результатами проведеного експериментального випробування, застосування розробленої методики значно скоротило час на розробку ігор. У середньому витрати часу при розробці однієї гри скоротилися вдвічі. Розробка самої системи та однієї гри на основі методики займає 97 днів, тоді як розробка такої системи з однією грою, але без використання методики займає 110 днів. Час розробки зменшився на 12%. При додаванні гри до системи час розробки гри скорочується на 47%.

Напрями подальшого дослідження включають додавання до системи нових навчальних комбінаторних ігор та реалізацію режиму гри, в якому користувачі системи могли б грати один проти одного.

#### Список літератури:

1. John H., Peters A. K. Conway. On Numbers and Games. CRC Press; 2nd edition. – December 11, 2000 – pp. 242
2. Berlekamp E., Conway J., Guy R. Winning ways for your mathematical plays. 2nd edition. – ISBN 1-56881-130-6 (alk.paper) – 2001, – pp. 297
3. Richard Guy. Fair Game: How to Play Impartial Combinatorial Games. Exploration in Math Series – 1989 – pp. 113
4. Thomas S., Ferguson A. Course in Game Theory – 2020 – pp. 46
5. Lucas Chess. An easy way to play and train chess on your PC. URL: <https://lucaschess.pythonanywhere.com/> (дата звернення 24.09.2023).
6. Martin Blume. Arena Chess GUI. URL: <http://www.playwitharena.de/> (дата звернення 24.09.2023).
7. Alexander Svirin. Aurora Borealis Draughts professional – URL: <http://aurora.draughtsworld.com/> (дата звернення 24.09.2023).
8. SmartGo One. Learn, play, study Go. URL: <https://smartgo.com/> (дата звернення 24.09.2023).
9. Gunnar Andersson. Wzebra, Othello game. URL: <http://radagast.se/othello/> (дата звернення 24.09.2023).

10. Рувінська В.М., Тройніна а.с. Інформаційна технологія (методика) розробки навчальних комбінаторних ігор: перші кроки / Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки». Том 34 (73) № 6, 2023 – с. 140-146.

11. Hunicke R., Leblanc M., Zubek R. MDA: A Formal Approach to Game Design and Game / Proceedings of the AAAI-04 Workshop on Challenges – URL: <https://www.cs.northwestern.edu/~hunicke/MDA.pdf> (дата звернення 24.09.2023).

12. Djaouti D., Alvarez J., Jessel J.-P., Methel G., Molinier P. A Gameplay Definition through Videogame Classification. International Journal of Computer Games Technology – URL: <http://dx.doi.org/10.1155/2008/470350>. (дата звернення 24.09.2023).

13. Jeremy Gibson Bond. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C# 3<sup>rd</sup>- 2022 – pp. 907

### **Ruvinska V.M., Troynina A.S. INFORMATION TECHNOLOGY FOR THE DEVELOPMENT OF TRAINING COMBINATOR GAMES**

*Nowadays, software systems are developed only for the most popular and complex combinatorial games, such as chess, checkers, etc. It is proposed to create this kind of system for simple impartial games, in particular "NIM", "NIM with restrictions" and "Dates", which allow users not only to play, but also to study the theory of combinatorial games. The main goal of the work is to reduce the time for creating educational combinatorial games, for this a methodology (IT) for their development has been created, which has four stages, such as requirements analysis, design, implementation and testing, as for any software system, but all of them have a generalized character, that is, they are suitable for the development of various combinatorial games. This article is the second in a series of two articles on this topic, the first detailed the early stages of the technology, and this one provides the final steps, i.e. the general MDA structure for the requirements analysis stage; a description of how it is necessary to perform each stage of the general methodology when developing a specific educational combinatorial game; creating specific combinatorial games, as well as analyzing whether the proposed technology significantly reduced the time for their development. As a result, a software system was implemented using the created methodology (IT), three simple combinatorial games were developed: "NIM", "NIM with restrictions", "Dates", which not only play with the user, but also teach how to play optimally, that is, help to determine winning and losing game states, giving hints on how to make a play properly, etc. Unity2D, NetBeans IDE for Java, git were used in the implementation of the games. It is shown that the time to create each of the games, thanks to the generalized methodology, was reduced by an average of two times compared to the development of each one separately.*

**Key words:** technology for development of educational combinatorial games, mathematical theory of combinatorial games, "NIM", "NIM with restrictions", "Dates".