

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра математики факультету інформатики

**ОПТИМАЛЬНІ СТРАТЕГІЇ В БАГАТОКРОКОВІЙ ГРІ ЗІ
СКІНЧЕННИМ ГОРИЗОНТОМ НА ПРИКЛАДІ ГРИ
«РЕНДЗІЮ»**

**Текстова частина до курсової роботи
за спеціальністю „Прикладна математика” 113**

Керівник курсової роботи
к.ф.-м.н., доц. Чорней Р.К.

“___” _____ 2020 р.
(підпис)

Виконав студент 4 курсу
факультету інформатики
Прокопенко В.Р.

“___” _____ 2020 р.

Кафедра математики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри математики,
проф., д.ф. –м.н. Олійник Б. В.

(підпис)

„_____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Прокопенку В. Р. факультету інформатики 4 курсу БП

ТЕМА Оптимальні стратегії в багатокроковій грі зі скінченним горизонтом на прикладі гри «рендзю»

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Анотація

Вступ

РОЗДІЛ 1: Основні теоретичні відомості про рендзю

РОЗДІЛ 2: Програмна реалізація гри «рендзю»

Висновки

Список використаної літератури

Дата видачі „_____” _____ 2020р. Керівник _____

(підпис)

Тема: Оптимальні стратегії в багатокроковій грі зі скінченним горизонтом на прикладі гри «рендзю»

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу
1.	Отримання завдання на дипломну роботу.	15.10.2019
2.	Огляд технічної літератури за темою роботи.	14.11.2019
3.	Аналіз існуючих досліджень за темою роботи.	12.12.2019
3.	Розробка програмного продукту.	11.01.2020
4.	Написання вступу та плану роботи.	05.03.2020
6.	Написання основних розділів роботи.	09.03.2020
7.	Створення слайдів для доповіді та написання доповіді.	25.03.2020
8.	Коригування роботи відповідно до вимог щодо оформлення робіт.	01.04.2020
9.	Остаточне оформлення пояснювальної роботи та слайдів.	03.04.2020
10.	Коригування роботи згідно із зауваженнями керівника.	07.04.2020
11.	Захист курсової роботи	19.04.2020

Студент Прокопенко В. Р.

Керівник Чорней Р.К.

“ _____ ”

ЗМІСТ

Анотація до курсової роботи	5
ВСТУП	6
РОЗДІЛ 1: ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО РЕНДЗЮ.....	7
1.1 Правила рендзю	7
1.2 Основні структури.....	9
1.3 Дебюти.....	12
1.4 Павза	13
1.4 Захист	14
РОЗДІЛ 2: ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ «РЕНДЗЮ»	16
2.1 Складання технічного завдання.....	16
2.2 Реалізація елементів інтерфейсу.....	16
2.3 Реалізація алгоритму гри.....	18
ВИСНОВКИ.....	21

Анотація до курсової роботи

В цій курсовій роботі було проаналізована теоретична інформація про багатокрокову гру «рендзю». Для закріплення теоретичної частини був реалізований програмний продукт. У реалізації було використано мову програмування Python та такі фреймворки: pygame – для роботи з 2D іграми; wxPython – для рендеру графічного інтерфейсу. Також використовувався графічний редактор Figma – для малювання пояснювальних рисунків та графічних елементів програмної реалізації.

ВСТУП

Актуальність теми: Світ інтелектуальних ігор дуже багатий. Майже всі знайомі з шахами і шашками, багато хто чув про го і бридж, але про рендзю, сянци, сьогодні знають вже далеко не всі. Тим часом і в маловідомих іграх нерідко можна знайти не менше краси і глибини, в тому числі і обчислювальної. Тому актуальною проблемою є необізнаність у різноманітні інтелектуальних ігор, що призводить до інтелектуальної деградації від покоління до покоління.

Мета і завдання дослідження: продемонструвати доцільність використання рендзю як однієї з варіантів багатокрокових інтелектуальних ігор.

Джерела для дослідження: використані іноземні джерела – наукова література та статті. Більш детально можна ознайомитися в розділі «Список використаних джерел».

Наукова новизна одержаних результатів: Підтвердження можливості програмної реалізації таких специфічних інтелектуальних ігор як рендзю.

У ХХ столітті в теорії ігор знайшлося місце для ігор « k в ряд на дошці m на n ». Цей клас ігор навіть отримав власну назву: m, n, k - гри. З точки зору класифікації вони є детермінованими іграми (як і го, але не як нарди), з повною інформацією (як і шахи, але не як бридж), з раптовою смертю (як і шахи, але не як го), дивергентними (як і го, але не як шахи). Ці терміни означають наступне: в грі повністю виключена випадковість; вся інформація про стан в грі і всіх можливих ходах гравців в кожен момент доступна кожному гравцеві; гра може закінчитися за умови досягнення одним з гравців певної позиції (мат в шахах, п'ятірка в рендзю); число наявних на дошці «фігур» i , головне, кількість можливих позицій зростає від дебюту до ендшпіль.

Зрозуміло, що про сильне рішення рендзю мова йти не може: число нулів в кількості можливих позицій в рендзю перевищує кількість нулів в, наприклад, можливих шахових позицій в два-три рази.

РОЗДІЛ 1: ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО РЕНДЗЮ

1.1 Правила рендзю

Гравець з чорними каменями починає гру. Перший хід повинен бути зроблений у центр дошки. Усі камені розміщуються на перетинах ліній замість того, щоб розміщуватися посередині квадратів. Після цього білий колір переходить на будь-яке вільне місце на дошці (див. рисунок 1). Щоб виграти, потрібно поставити п'ять каменів в лінію за будь-яким напрямком: горизонтальна, вертикальна або ж діагональна «п'ятірка».

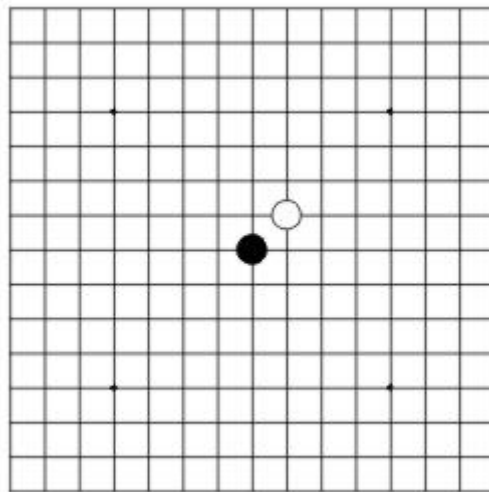


Рисунок 1 – Початок гри

За допомогою цих правил ми б змогли грати в рендзю 100 років тому. Проте зараз цього недостатньо, оскільки японські професійні гравці помітили, що чорні завжди мали перевагу над білими. Необхідно було встановити додаткові правила, щоб збалансувати гру та дати гравцю білих більше шансів на опір.

Отже, ось три фоли, які обмежують чорних:

- 1) Чорному гравцю заборонено робити вилку (хід, який створює одночасно дві атакуючі лінії) 3x3.
- 2) Чорному гравцю заборонено робити вилку 4x4.

3) Чорному гравцеві заборонено робити ряд з шести і більше каменів поспіль.

Ці обмеження стосуються лише чорної сторони. Білі ж мають розв'язані руки. Як тоді чорні можуть виграти гру? Відповідь – єдиний дозволений виграш для чорних – це вилки 4x3 (див. рисунок 2).

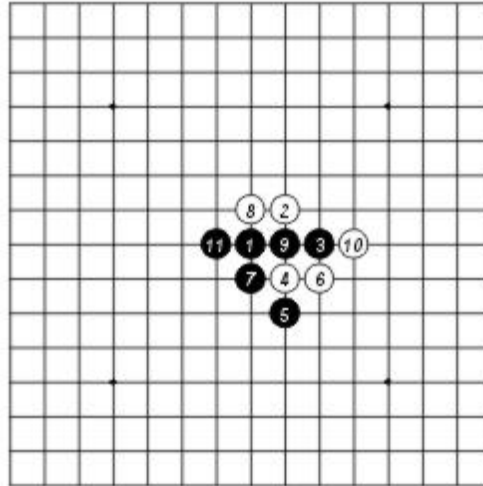


Рисунок 2 – Чорні мають вилку 4x3

Аспект заборон привносить абсолютно новий тактичний інструмент у рендзю – використання фолів задля перемоги. Цей інструмент є зброєю лише для білих гравців (див. рисунок 3). Білий гравець використовує заборонені 3x3 вилки і, отже, виграє гру. Оскільки чорному не дозволено грати до точки X, чорний повинен грати в іншому місці, а білі можуть виграти на наступному кроці, зробивши п'ять поспіль у точці X.

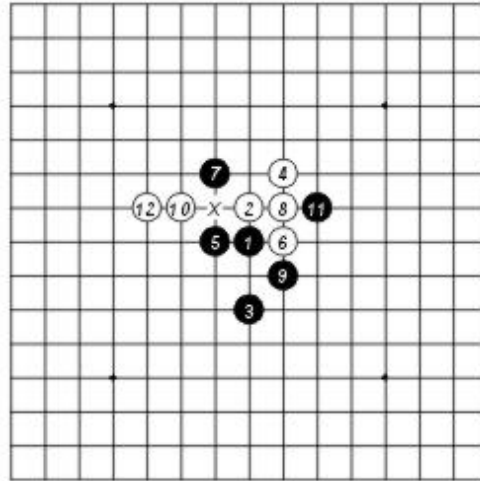


Рисунок 3 – Використання фола білими

Нічия в партії буде в тому випадку, коли суперники вважають, що можливість іншого результату виключена (наприклад, вся дошка заставлена каменями). Будь-який з суперників в якості чергового ходу може відмовитися виставити камінь на дошку. При відмові обома суперниками, партія закінчується внічию.

1.2 Основні структури

Ми витрачаємо багато часу, щоб побудувати ряд, і коли ми вже майже зробили п'ятірку, наш суперник просто перекриває нам шлях. Щоб досягти успіху у цій грі, ми повинні зрозуміти деякі основні структури, які можуть бути дуже корисними в атаці. Вміле використання таких структур допоможе нам виграти гру. Наша зброя – це трійки та четвірки. Якщо противник їх ігнорує, а не блокує – ми переможемо.

Четвірки – лінія, яка загрожує стати п'ятими поспіль наступним кроком.

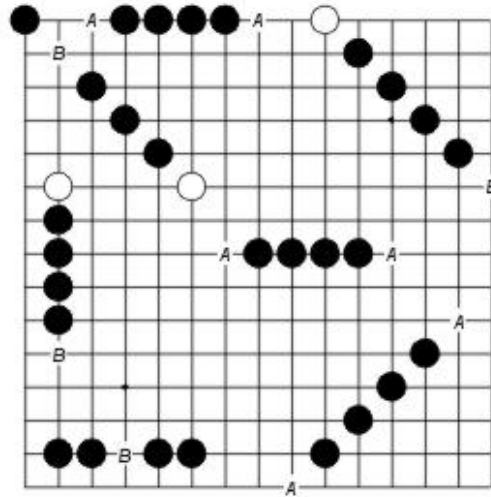


Рисунок 4 – Приклади четвірок

Чорні можуть поставити п'ять поспіль з точок А або В (див. рисунок 4). Четвірки з позначкою "В" - це напіввідкриті четвірки. Якщо хід білих в грі, вони здатні захиститись, поставивши білий камінь у точку В. Чотири в ряд, позначені символом "А", – це відкриті четвірки.

Це типова ситуація з виграшем у рендзю. Кожного разу, коли гравцеві вдасться отримати відкриту четвірку, захиститись неможливо, тому що обидва кінці четвірки є відкритими, і незалежно від того, який кінець з чотирьох каменів блокується опонентом, другий кінець залишатиметься відкритим, і п'ятірка все одно буде поставлена.

Отже, головною тактикою гравця має бути створення відкритої четвірки. Відкрита четвірка може бути створена лише з відкритої трійки. Мати відкриту четвірку практично означає перемогу, тому суперник повинен бути дуже обережним, коли є відкрита трійка на дошці.

Дуже важливо вказати різницю між справжньою відкритою трійкою та псевдотрійкою, яка виглядає відкритою, але насправді це не так. Багато початківців поспішають перекрити трійки, які не були відкритими, і захищати їх насправді не було потреби. Ми бачимо приклади псевдотрійок (див. рисунок 5), які є безпечними для білого гравця.

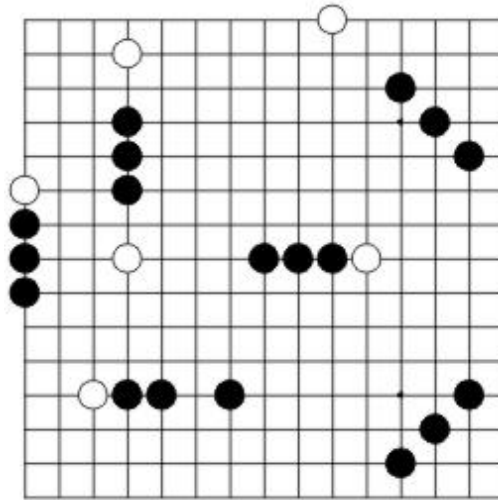


Рисунок 5 – Приклади псевдотрійок

Тепер ми зрозуміли значення відкритої трійки та четвірки. Вони є потужним інструментом нападу. Однак, якщо ми просто будемо ставити по черзі ці структури, нам ніколи не вдасться виграти. Мудрий супротивник блокує всі наші відкриті трійки та четвірки, і в нас немає шансів перемогти.

Для того, щоб перемогти більш вмілих гравців, потрібно використовувати більш складні методи атаки. Простіше кажучи, нам слід спробувати створити вилку. Якщо ми це зробимо, тоді в суперника виникають великі неприємності, оскільки він не встигає заблокувати їх обох, і одна з ліній потім залишиться відкритою, і наша гра вже майже дійшла до тріумфу. Вилки 4x4 утворюються поставленим каменем у точки F та G (див. рисунок 6), а вилки 3x3 – у точки K та J.

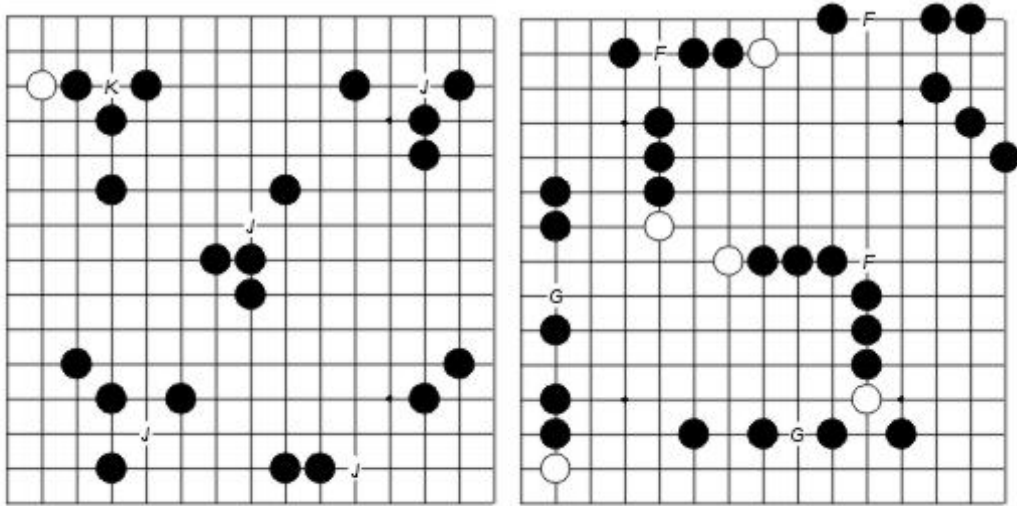


Рисунок 6 – Вилки 3x3 та 4x4

1.3 Дебюти

Дебют - початок гри, перші п'ять-десять ходів. У рендзю дебют має найбільше значення, адже помилка на початку має катастрофічні наслідки в середині гри та різко підвищує шанс програшу. Усього існує 26 канонічних дебютів. Вони поділяються на вертикальні та діагональні (див. рисунок 7).

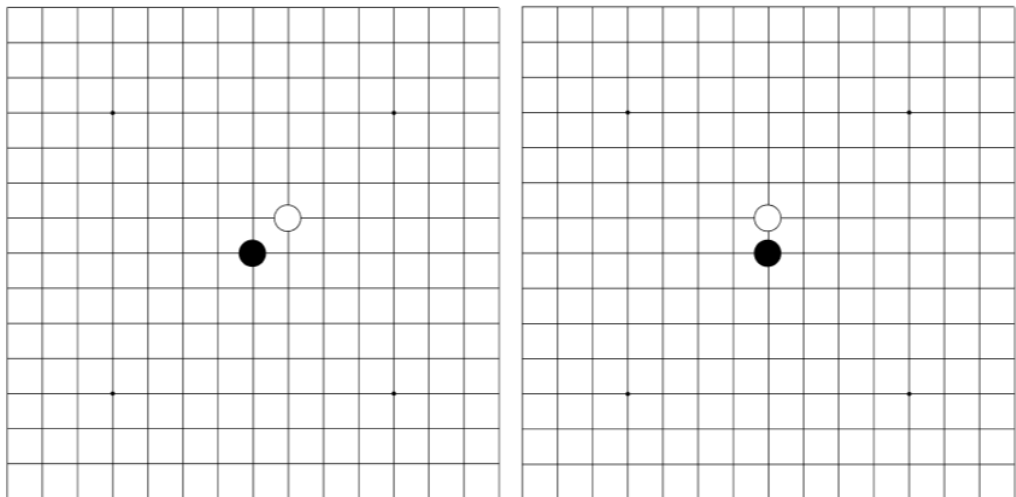


Рисунок 7 – Вертикальні та діагональні дебюти

Обираючи той чи інший дебютний варіант, гравець повинен уникати механічного розігрування початку партії. Слід привчати себе до творчого

осмислення гри в дебюті. В чому ж полягає таке осмислення? Зазвичай, гравець прагне до створення на гральній дошці мітельшпільних позицій, відповідних до його особистих смаків та стилю гри. Саме тому вже у стадії дебюту він повинен намітити план дій, який допоможе йому отримати бажану розстановку сил в середині гри. Звісно, в рендзю протистоять дві сили. Саме тому, поряд з основним планом, повинні бути запасні варіанти на той випадок, якщо суперник вважатиме за краще ухилитися від даного дебюту.

1.4 Павза

Павза - хід, що підсилює позицію гравця. Застосовувати будь-яку павзу можна тільки тоді, коли у суперника немає можливості закінчити гру серією атак. Один з видів - свої камені ставляться далеко від каменів суперника і тим самим створюється чисельна перевага на даному фланзі. Загалом, павза найбільш ефективна, коли з'єднує різні фланги потенційної атаки. Наприклад, чорні ставлять павзу у точку 1 (див. рисунок 8). В даній позиції цей блискучий хід призводить до того, що створюються можливості для атаки на двох різних флангах.

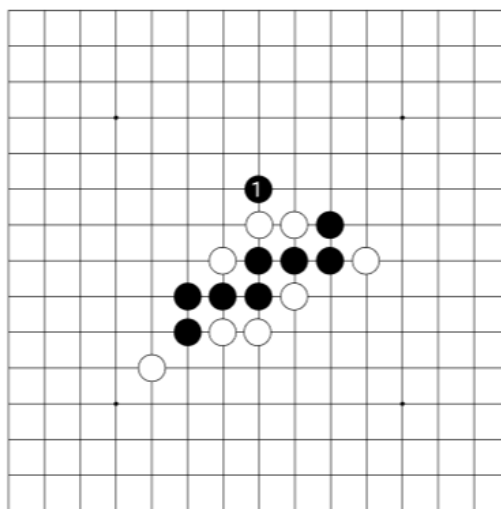


Рисунок 8 – Павза

1.4 Захист

Майже всі перераховані раніше прийоми в залежності від ситуації можуть мати або захисний, або атакуючий характер. Крім них є ще один, винятково захисний, прийом - побудова сітки. Сітка - інструмент глухого захисту, але після її побудови гравець в два ходи легко може перейти до активної гри. Хоча сітка (або її фрагменти) - дуже простий і часто ефективний засіб захисту, не потрібно захоплюватися її створенням, тому що суперник може легко зрозуміти наш задум і зірве його, а ми залишимося без жодних шансів на атаку. Наприклад, білі почали створювати свою захисну сітку (див. рисунок 9), яка може бути добудована їх ходами в пункти А.

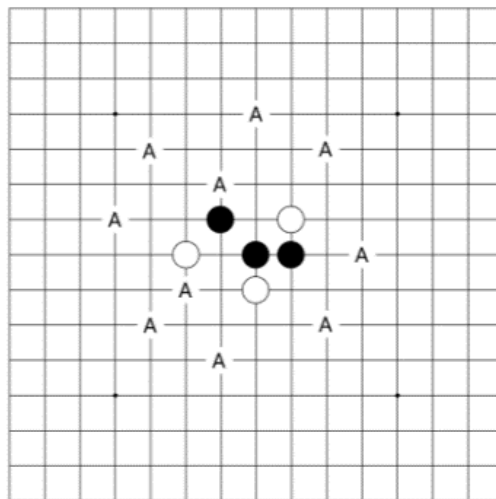


Рисунок 9 – Варіанти побудови сітки

Також, беручи до уваги правила рендзю, слід зазначити про захисні засоби білих за допомогою фолів. Часто білі можуть позбавити чорних неминучого виграшу, створюючи їм фол в етапі побудови переможної вишки 4x3. Наприклад, (див. рисунок 10) чорні в пункті А мають виграшну вишку 4x3. Побудовою атаки 1, білі повністю нейтралізують цю загрозу (по горизонталі може утворитися лише довгий ряд), а потім кроком 3 самі переходять в атаку, одночасно прикриваючи діагональну трійку чорних.

РОЗДІЛ 2: ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ «РЕНДЗІЮ»

2.1 Складання технічного завдання

Ми будемо створювати рендзю з режимом гри «гравець проти гравця». Цей режим працюватиме за наступним алгоритмом: спочатку обидва гравці вводять свої імена (це потрібно для декларування переможця), далі відображається гральна дошка, на якій вже знаходиться чорний камінь по центру, згідно з правилами. Опоненти ходять по черзі, аж поки програма не знайшла комбінацію з п'ятирок каменів. Після успішної перевірки з'являється вікно з привітанням ліпшого гравця та три кнопки. За те, щоб розпочати гру знову, не використовуючи ніяких маніпуляцій з іменами, буде відповідати кнопка «Грати ще». Кнопка «Дошка» потрібна задля перегляду кінця партії, аби суперники могли проаналізувати свої помилки та покращити навички. Нарешті, остання кнопка «Меню» повертає користувачів до головного меню, де можливо почати гру, змінивши імена опонентів.

Як зазначено в анотації до курсової роботи, ми обрали задля написання рендзю мову програмування Python. Цей вибір було зроблено через його міжплатформову реалізацію та сильну спільноту, яке створило безліч корисних бібліотек. Ми використаємо бібліотеку pygame та wxPython. Перший інструмент є перевірений часом для створення 2D ігор, а другий – лаконічне та просте в освоєнні знаряддя для створення графічного інтерфейсу. Також ми скористалися графічним редактором Figma – малювали елементи інтерфейсу.

2.2 Реалізація елементів інтерфейсу

Програмну реалізацію наведено у додатку А.

Кожна гра починається з головного меню. З безлічі варіантів графічного інтерфейсу ми вибрали найоптимальніший - програма відображує готове

зображення, створене перед цим у графічному редакторі. Figma має достатній функціонал для такої задачі – нею ми і скористалися.

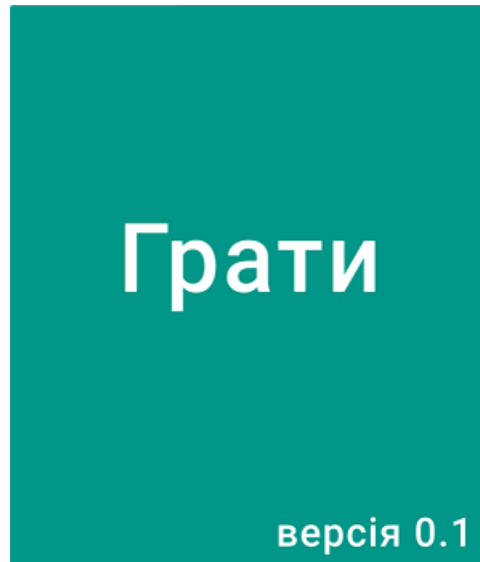


Рисунок 11 – Головне меню

Головне меню має сучасний дизайн в стилі «Material» (див. рисунок 11) та відповідає усім нашим вимогам. Наступним пунктом є реєстрація гравців. Тут скористаємося викликом діалогового вікна для обох сторін (див. рисунок 12). У заголовку підпишемо нумерацію гравців. Також, згідно з правилами рендзю, присвоїмо чорні камені першому гравцю, а білі – другому. Ці заходи направлені на перешкоджання плутанині під час гри.

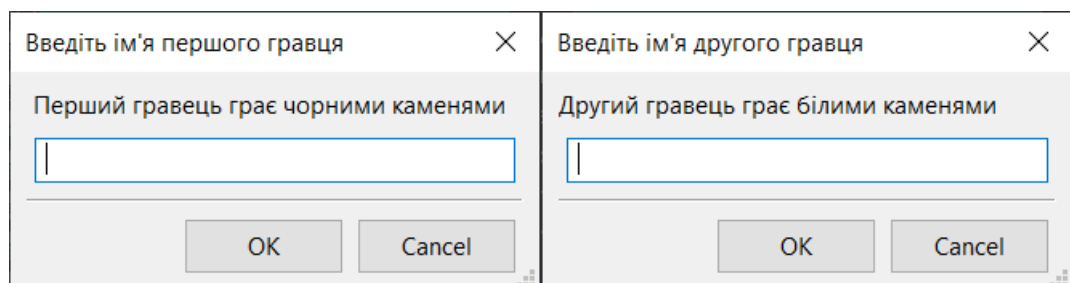


Рисунок 12 – Реєстрація гравців

2.3 Реалізація алгоритму гри

Програмну реалізацію наведено у додатку А.

Повторимо алгоритм створення головного меню: намалюємо в редакторі гральну дошку, чорні та білі камені. Після того як суперники ввели свої імена, програма створює нове вікно поверх попереднього. У ньому міститься зображення гральної дошки (див. рисунок 13) та вже поставленого чорного каменя по центру. Цей метод було впроваджено, щоб гравці не відволікалися кожну нову партію на обов'язкові правила початку гри, а планували свою переможню стратегію. Заголовок вікна показує імена суперників.

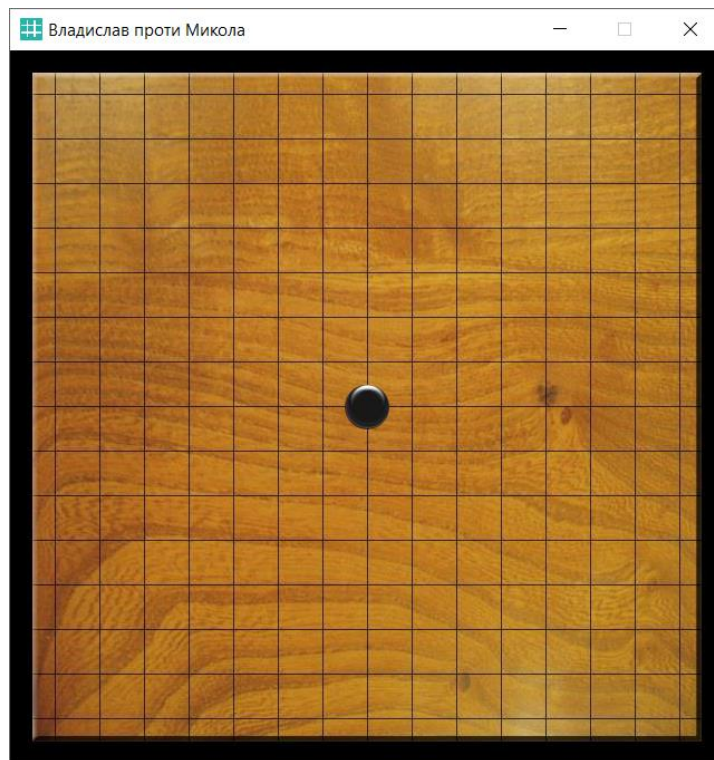


Рисунок 13 – Гральна дошка

Гра йде шляхом чергування ходів чорними та білими каменями. Його було реалізовано шляхом циклу, у якому перевіряється число на парність та додаванням одиниці до цього числа кожну ітерацію. Однією з проблем є правильне розташування каменів при клацанні мишкою на потрібне місце.

Зафіксуємо довжину та ширину зображення гральної дошки на рівні 640x640 пікселів. Зважаючи на борти шириною в 20 пікселів, ми маємо гральну зону з розмірами 600x600 пікселів. За правилами гри, дошка має бути розлінована 15 вертикальними і 15 горизонтальними лініями. Поділивши 600 на 15 ми отримали 40 пікселів – квадрат для одного каменя. Кожні 40 пікселів починається новий квадрат. При клацанні мишкою на дошку програма дізнається координати курсору та перевіряє її з позицією кожного з 225 квадратів, поки позиція не ввійде в проміжок координат потрібного квадрату. У потрібний квадрат ставиться камінь (див. рисунок 14). Зайнятий квадрат заноситься до списку, щоб не можна було поставити інший камінь на його місце.

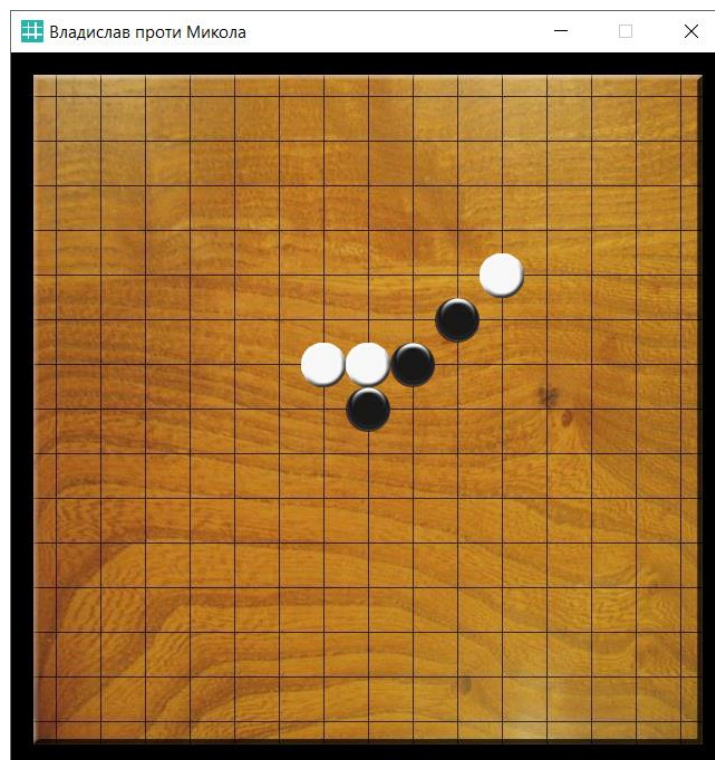


Рисунок 14 – Вигляд дошки під час гри

Гра триває, проте без алгоритму визначення переможця вона буде тривати поки не закінчиться вільне місце. Після кожного ходу береться останній зайнятий квадрат та перевіряється зайнятість іншими квадратами в

усіх напрямках чотирма однаковими з останнім каменем. Це не дуже швидкий варіант, проте простий та ефективний. Після успішної перевірки вікно з гральною дошкою закривається. Замість нього з'являється вікно (див. рисунок 14) з декларуванням переможця та можливістю переглянути дошку, зіграти ще раз під тими ж іменами та повернутися до головного меню.

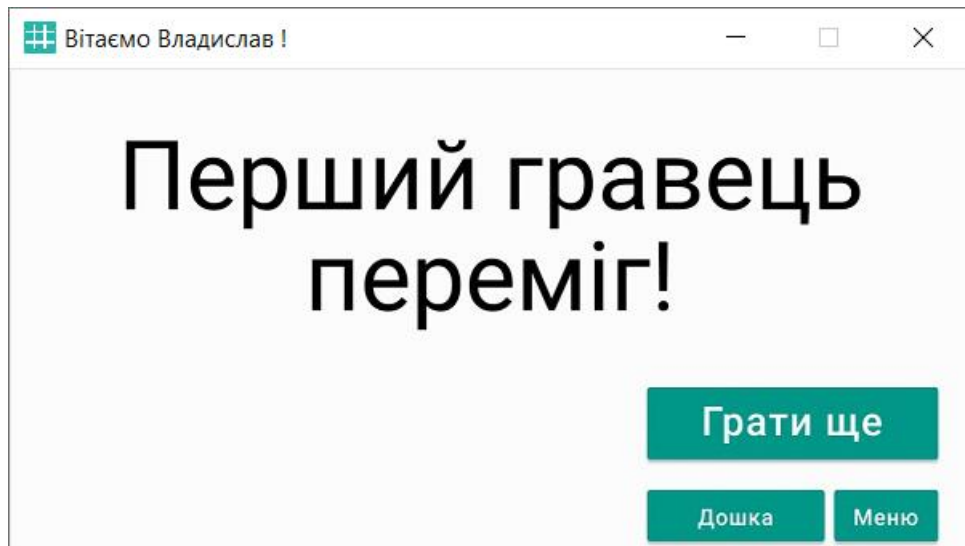


Рисунок 15 – Декларування переможця

ВИСНОВКИ

В цій курсовій роботі було розглянуто

- Основні теоретичні відомості про багатокрокову гру зі скінченим горизонтом «рендзю».
- Програмну реалізацію рендзю.

Наразі «низка перлів» є лише грою для тих, хто не зміг знайти себе в шахах, проте очікується збільшення зацікавленості населення через глобалізацію. Тому ми маємо очікувати, що задля того, щоб гравці нашої країни мали достатній рівень теоретичної та практичної підготовки під час міжнародних змагань, потрібно буде мати навички майбутнім тренерам та секціям у цілому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is Renju? [Електронний ресурс] / Aivo Oll// — 2006. — Режим доступу до ресурсу: <http://www.renju.net/study/rules.php>
2. Сложнее шахмат [Електронний ресурс] / Д. Епифанов // — 2016. Режим доступу до ресурсу : <https://nplus1.ru/material/2016/09/27/renjugame>
3. Рендзю для начинающих [Електронний ресурс] / А. Носовский, А. Сокольский // — 2003. — Режим доступу до ресурсу: http://www.piskvorky.net/cms/uploads/files/Rendzyu_Dlya_Nachinayuschikh.pdf
4. И. Кан От дебюта к миттельшпилю / И. Кан. — М. : «Физкультура и спорт», 1978. – 136с.
5. A Way to Improvement of player's standard (Part 1) [Електронний ресурс] / Alexander Nosovsky // — 2001. — Режим доступу до ресурсу: <http://www.vcpr.cz/files/download/awaytoimprovementofplayer.pdf>

ДОДАТКИ

Додаток А
(обов'язковий)
Текст програми “Гра рендзю”

```

1  import wx
2  import pygame, sys
3  from pygame.locals import *
4
5  class renju(wx.Frame):
6      def __init__(self, parent, id):
7          wx.Frame.__init__(self, parent, id, 'Рендзю', size_=(310,390))
8          panel = wx.Panel(self)
9          pic = wx.Image("twoplayer.bmp", wx.BITMAP_TYPE_BMP).ConvertToBitmap()
10         self.button = wx.BitmapButton(panel, -1, pic, pos_=(-5, -5))
11         self.Bind(wx.EVT_BUTTON, self.dome, self.button)
12         self.button.SetDefault()
13
14         def dome(self, event):
15             textbox = wx.TextEntryDialog(None, "Перший гравець грає чорними каменями", "Введіть ім'я першого гравця")
16             if textbox.ShowModal() == wx.ID_OK:
17                 Name1 = textbox.GetValue()
18                 textbox = wx.TextEntryDialog(None, "Другий гравець грає білими каменями", "Введіть ім'я другого гравця")
19                 if textbox.ShowModal() == wx.ID_OK:
20                     Name2 = textbox.GetValue()
21                 else:
22                     return
23             else:
24                 return
25             renju.twoplayer(self, event, Name1, Name2)
26
27         def twoplayer(self, event, Name1, Name2):
28             fread = open('theme.txt', 'r')
29             default = fread.readline()
30             fread.close()
31             bif = default
32             bsif = "whites.png"
33             wsif = "blacks.png"
34             i_icon = "icon.png"
35             pygame.init()
36             icon = pygame.image.load(i_icon)
37             pygame.display.set_icon(icon)
38             screen = pygame.display.set_mode((640,640),0,32)
39             background = pygame.image.load(bif).convert()
40             black = pygame.image.load(bsif).convert_alpha()
41             white = pygame.image.load(wsif).convert_alpha()
42             pygame.display.set_caption(str(Name1)+' проти '+str(Name2))
43             blacks = []
44             whites = []
45             count = 1
46             blacks.append((300,300))
47             screen.blit(background,(0,0))
48             screen.blit(black,(300,300))
49             pygame.display.update()
50

```

```

51 while True:
52     pos = [0,0]
53     for event in pygame.event.get():
54         if event.type == QUIT:
55             pygame.quit()
56
57         if event.type == MOUSEBUTTONDOWN:
58             pos = list(event.pos)
59             flag = 1
60             x = 20
61             while x<620:
62                 if pos[0]>=x and pos[0]<x+40:
63                     pos[0] = x
64                     break
65                 x = x+40
66             y = 20
67             while y<620:
68                 if pos[1]>=y and pos[1]<y+40:
69                     pos[1] = y
70                     break
71                 y = y+40
72             j = 0
73             while j < len(whites):
74                 if pos[0] == whites[j][0] and pos[1] == whites[j][1]:
75                     flag = 0
76                     break
77                 j = j+1
78             j = 0
79             while j < len(blacks):
80                 if pos[0] == blacks[j][0] and pos[1] == blacks[j][1]:
81                     flag = 0
82                     break
83                 j = j+1
84             if pos[0]>620 or pos[0]<20 or pos[1]>620 or pos[1]<20:
85                 flag=2
86
87             if flag == 1:
88                 count = count+1
89                 if count%2 == 1:
90
91                     blacks.append((pos[0],pos[1]))
92
93                 elif count%2 == 0:
94
95                     whites.append((pos[0],pos[1]))
96
97             i = 1
98             while i <= count:
99                 if i%2 == 0:
100                     screen.blit(white,whites[i//2 -1])
101                     pygame.display.update()
102                 else:
103                     screen.blit(black,blacks[(i-1)//2_])
104                     pygame.display.update()
105                 i = i+1
106
107             stone = ""
108             turn = []
109             flag = 0
110             if count % 2 == 1:
111                 stone = Name1
112                 turn = blacks
113             elif count % 2 == 0:

```

Open E

```

114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177

    stone = Name2
    turn = whites
I = 0
while I < len(turn):
    a = (turn[I][0],turn[I][1])
    n = 1
    while n < 5:
        if (a[0]+40*n, a[1])in turn:
            n = n+1
        else:
            break
    if n == 5:
        b = (a[0]+40*5, a[1])
        flag = 1
        break
    n = 1
    while n < 5:
        if (a[0]+40*n, a[1]+40*n)in turn:
            n = n+1
        else:
            break
    if n == 5:
        flag = 1
        b = (a[0]+40*5, a[1]+40*5)
        break
    n = 1
    while n < 5:
        if (a[0]+40*n, a[1]-40*n)in turn:
            n = n+1
        else:
            break
    if n == 5:
        b = (a[0]+40*5, a[1]-40*5)
        flag = 1
        break
    n = 1
    while n < 5:
        if (a[0], a[1]+40*n)in turn:
            n = n+1
        else:
            break
    if n == 5:
        b = (a[0], a[1]+40*5)
        flag = 1
        break
    I = I+1
if flag == 1:
    pygame.time.delay(1000)
    pygame.quit()
    i_icon = 'icon.png'
    if turn == blacks:
        bif1 = 'win1.jpg'
    if turn == whites:
        bif1 = 'win2.jpg'
    pygame.init()
    pygame.display.set_icon(icon)
    pygame.display.set_caption('Битаямо '+stone+' !')
    screen = pygame.display.set_mode((600,300),0,32)
    background = pygame.image.load(bif1).convert()
    screen.blit(background,(0,0))
    pygame.display.update()
    while True:
        for event in pygame.event.get():
            if event.type == QUIT:

```

```

178         pygame.quit()
179         if event.type == MOUSEBUTTONDOWN:
180             pos = list(event.pos)
181             if pos[0] > 400 and pos[0] < 580 and pos[1] > 200 and pos[1] < 240:
182                 pygame.quit()
183                 renju.twoply(self,event,Name1,Name2)
184             if pos[0] > 520 and pos[0] < 580 and pos[1] > 270 and pos[1] < 290:
185                 pygame.quit()
186             if pos[0] > 390 and pos[0] < 500 and pos[1] > 270 and pos[1] < 290:
187                 pygame.quit()
188                 renju.viewbrd(self,event,blacks,whites)
189         screen.blit(background,(0,0))
190     screen.blit(background,(0,0))
191
192
193     def viewbrd(self, event, blacks, whites):
194         fread = open('theme.txt', 'r')
195         default = fread.readline()
196         fread.close()
197         bif = default
198         bsif = "whites.png"
199         wsif = "blacks.png"
200         i_icon = "icon.png"
201         pygame.init()
202         icon = pygame.image.load(i_icon)
203         pygame.display.set_icon(icon)
204         screen = pygame.display.set_mode((640,640),0,32)
205         background = pygame.image.load(bif).convert()
206         black = pygame.image.load(bsif).convert_alpha()
207         white = pygame.image.load(wsif).convert_alpha()
208         pygame.display.set_caption('Останній вигляд дошки')
209         screen.blit(background,(0,0))
210         pygame.display.update()
211         i = 0
212         while i < len(whites):
213             screen.blit(white, whites[i]...)
214             pygame.display.update()
215             i = i+1
216         i = 0
217         while i < len(blacks):
218             screen.blit(black, blacks[i]...)
219             pygame.display.update()
220             i = i+1
221         while True:
222             for event in pygame.event.get():
223                 if event.type == QUIT:
224                     pygame.quit()
225             screen.blit(background,(0,0))
226
227     if __name__ == '__main__':
228         app = wx.App(False)
229         frame =renju(parent = None, id = -1)
230         frame.Show()
231         app.MainLoop()

```