

**ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN-ĐIỆN TỬ**

ĐỒ ÁN THIẾT KẾ I

Thiết kế và Xây dựng Trò chơi Othello bằng ngôn ngữ lập trình Java

TRỊNH LONG VŨ

Vu.TL223831@sis.hust.edu.vn

Chuyên ngành điện tử – viễn thông

Giảng viên hướng dẫn: PGS. TS. Vương Hoàng Nam

HÀ NỘI, 7/2025

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án, em đã nhận được sự giúp đỡ và định hướng quý báu từ giảng viên và sự động viên từ gia đình. Em xin gửi lời cảm ơn chân thành đến thầy PGS. TS. Vương Hoàng Nam đã đưa ra định hướng ban đầu để em xác định được cách triển khai đồ án phù hợp và rõ ràng.

Em cũng xin chân thành cảm ơn các thầy cô trong khoa đã truyền đạt cho em những kiến thức nền tảng trong suốt quá trình học tập tại trường, giúp em có cơ sở để hoàn thành đồ án này.

Em đặc biệt biết ơn gia đình vì đã luôn ủng hộ, tạo điều kiện thuận lợi để em có thể tập trung học tập và làm việc nghiêm túc trong suốt thời gian qua.

Do giới hạn về thời gian và kiến thức chuyên môn, đồ án không tránh khỏi những thiếu sót. Em rất mong nhận được các ý kiến đóng góp để hoàn thiện hơn trong các dự án sau này.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Đồ án "Thiết kế và Xây dựng Trò chơi Othello bằng ngôn ngữ lập trình Java" nhằm xây dựng một ứng dụng trò chơi Othello có giao diện đồ họa trực quan, cho phép người chơi đấu với nhau hoặc với máy tính. Đồ án áp dụng phương pháp lập trình hướng đối tượng (OOP) kết hợp mô hình thiết kế phần mềm MVC, với các thành phần như quản lý logic trò chơi, giao diện người dùng và trí tuệ nhân tạo đơn giản. Công cụ sử dụng gồm: Java JDK 14, thư viện Java Swing, môi trường phát triển VS Code, hệ thống quản lý mã nguồn Git/GitHub. Kết quả đạt được đáp ứng đầy đủ các yêu cầu chức năng và phi chức năng đã đề ra, đảm bảo tính ổn định, hiệu năng và tương tác mượt mà. Đồ án có tính thực tế cao, có thể mở rộng bằng cách nâng cấp AI (Minimax, Alpha-Beta), cải thiện giao diện, thêm chế độ chơi trực tuyến hoặc tính năng lưu ván chơi. Qua dự án, sinh viên đã củng cố vững chắc kiến thức OOP, phát triển kỹ năng thiết kế hệ thống, xây dựng giao diện Java, xử lý logic trò chơi và tư duy thuật toán cơ bản..

Sinh viên thực hiện
Ký và ghi rõ họ tên

Vũ
Trình Long Vũ

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU VÀ PHÂN TÍCH YÊU CẦU	7
1.1 Giới thiệu về trò chơi Othello	7
1.1.1 Tổng quan.....	7
1.1.2 Luật chơi cơ bản.....	7
1.1.3 Cách chơi chi tiết	8
1.2 Mục tiêu của đồ án	8
1.2.1 Mục tiêu chính.....	8
1.2.2 Mục tiêu học tập.....	8
1.3 Yêu cầu chức năng.....	8
1.4 Yêu cầu phi chức năng.....	9
1.4.1 Yêu cầu về hiệu năng	10
1.4.2 Yêu cầu về giao diện	10
1.4.3 Yêu cầu về tính ổn định	10
1.4.4 Yêu cầu về khả năng sử dụng	10
1.5 Công cụ sử dụng	10
1.5.1 Ngôn ngữ lập trình	10
1.5.2 Môi trường phát triển	10
CHƯƠNG 2. THIẾT KẾ HỆ THỐNG	11
2.1 Kiến trúc tổng thể.....	11
2.1.1 Giao diện người dùng (UI - User Interface)	11
2.1.2 Quản lý trò chơi (Game logic)	11
2.1.3 Trí tuệ nhân tạo đơn giản (Simple AI).....	11
2.2 Sơ đồ trạng thái (State diagram)	11
2.3 Sơ đồ hoạt động (FlowChart).....	12
2.4 Sơ đồ lớp (Class Diagram).....	14
2.4.1 GamePanel	14
2.4.2 GridGame	15
2.4.3 GridCell.....	15
2.4.4 Position.....	15
2.4.5 Rectangle.....	16

2.4.6	SimpleAI	16
2.4.7	Giao diện và Lớp cơ sở (Interfaces and Base Classes):.....	16
CHƯƠNG 3. TRIỂN KHAI CÀI ĐẶT TRÒ CHƠI.....		17
3.1	Các chức năng đã triển khai.....	17
3.2	Giao diện trò chơi.....	17
3.2.1	Giao diện chọn chế độ chơi.....	17
3.2.2	Giao diện bàn cờ	18
3.3	Code hoàn chỉnh.....	20
CHƯƠNG 4. KIỂM THỬ		21
4.1	Kiểm thử chức năng	21
4.2	Kiểm thử phi chức năng.....	22
CHƯƠNG 5. KẾT LUẬN.....		24
5.1	Những gì đã học được.....	24
5.2	Định hướng phát triển	24

DANH MỤC HÌNH VẼ

Hình 2-1. Sơ đồ phân cấp chức năng	11
Hình 2-2. Sơ đồ trạng thái của trò chơi Othello.....	12
Hình 2-3. Sơ đồ hoạt động của trò chơi Othello	13
Hình 2-4. Sơ đồ lớp của trò chơi Othello.....	14
Hình 3-1. Giao diện chọn chế độ chơi	18
Hình 3-2. Giao diện bàn cờ 1	18
Hình 3-3. Giao diện bàn cờ 2	18
Hình 3-4. Giao diện bàn cờ 3	19
Hình 3-5. Giao diện bàn cờ 4	19
Hình 3-6. Giao diện bàn cờ 5	20

DANH MỤC BẢNG BIỂU

Bảng 1-1. Bảng yêu cầu chức năng của trò chơi Othello.....	9
Bảng 3-1. Bảng chức năng đã triển khai	17
Bảng 4-1. Bảng kiểm thử chức năng.....	21
Bảng 4-2. Bảng kiểm thử phi chức năng.....	23

CHƯƠNG 1. GIỚI THIỆU VÀ PHÂN TÍCH YÊU CẦU

1.1 Giới thiệu về trò chơi Othello

1.1.1 Tổng quan

Othello (hay còn được biết đến với tên gọi khác là Reversi) là một trò chơi chiến thuật cổ điển dành cho hai người chơi, có lịch sử phát triển lâu đời. Trò chơi này được phát minh vào năm 1883 bởi Lewis Waterman và đã nhanh chóng trở thành một trong những trò chơi bảng (board game) phổ biến nhất trên toàn cầu. Trò chơi không chỉ đòi hỏi sự suy luận logic mà còn cần khả năng tư duy chiến lược và tiên đoán nước đi của đối thủ.

Othello có luật chơi tương đối đơn giản, dễ tiếp cận nhưng rất khó để thành thạo. Trò chơi rèn luyện tư duy phản xạ, khả năng phân tích tình huống và đưa ra quyết định tối ưu trong thời gian ngắn. Với những ưu điểm đó, Othello đã trở thành một trò chơi lý tưởng trong các cuộc thi trí tuệ, câu lạc bộ chiến lược và môi trường giáo dục.

1.1.2 Luật chơi cơ bản

Luật chơi Othello đơn giản nhưng đầy tính chiến thuật. Dưới đây là những nguyên tắc chính:

- Bàn cờ: Bàn cờ hình vuông kích thước 8x8, gồm 64 ô vuông bằng nhau.
- Quân cờ: Mỗi quân cờ có hai mặt – một mặt đen và một mặt trắng. Quân cờ sẽ được lật để thể hiện sự thay đổi người kiểm soát.
- Người chơi: Có hai người chơi – một người điều khiển quân đen (Black), người còn lại điều khiển quân trắng (White).
- Khởi đầu: Bốn quân cờ được đặt ở trung tâm bàn cờ theo hình dấu cộng (criss-cross): hai quân đen và hai quân trắng chéo nhau.
- Luật đặt quân: Mỗi lượt, người chơi đặt một quân cờ của mình lên bàn cờ sao cho kẹp được ít nhất một quân cờ của đối phương theo một đường thẳng (ngang, dọc hoặc chéo).
- Lật quân: Các quân cờ của đối phương bị kẹp giữa quân cờ vừa đặt và quân cờ cùng màu trước đó sẽ bị lật đổi thành màu của người chơi hiện tại.
- Luân phiên: Người chơi quân đen sẽ đi trước. Sau đó luân phiên từng lượt giữa hai người.
- Kết thúc trò chơi: Trò chơi kết thúc khi không còn nước đi hợp lệ nào cho cả hai người hoặc bàn cờ đã đầy.
- Thắng – Thua: Người có số lượng quân cờ nhiều hơn trên bàn cờ khi kết thúc trò chơi sẽ là người chiến thắng. Nếu số quân bằng nhau thì kết quả là hòa.

1.1.3 Cách chơi chi tiết

1. Bắt đầu: Bốn quân cờ được đặt giữa bàn cờ theo hình chéo đen - trắng.
2. Lượt đi đầu tiên: Người chơi quân đen luôn là người đi trước.
3. Tìm nước đi hợp lệ: Trước mỗi lượt, hệ thống xác định các ô có thể đặt quân hợp lệ – tức là có thể kẹp ít nhất một quân của đối thủ.
4. Thực hiện nước đi: Người chơi chọn một ô trong các ô hợp lệ, đặt quân vào đó.
5. Lật quân: Các quân cờ của đối thủ bị kẹp giữa sẽ bị lật đổi màu.
6. Chuyển lượt: Hệ thống tự động chuyển lượt cho người chơi còn lại. Nếu không có nước đi nào hợp lệ, lượt đó sẽ bị bỏ qua.
7. Kết thúc: Khi bàn cờ đầy hoặc cả hai người chơi đều không còn nước đi, trò chơi kết thúc.
8. Xác định kết quả: Người có nhiều quân cờ hơn sẽ giành chiến thắng.

1.2 Mục tiêu của đồ án

1.2.1 Mục tiêu chính

Mục tiêu của đồ án là xây dựng một ứng dụng trò chơi Othello đầy đủ tính năng với giao diện đồ họa trực quan và thân thiện, hoạt động ổn định trên nền tảng Java. Ứng dụng cần đáp ứng các yêu cầu:

- Cho phép người dùng chơi với nhau trên cùng một máy (Player vs Player – PvP).
- Cho phép người dùng chơi với máy tính sử dụng thuật toán đơn giản (Player vs AI).
- Cung cấp trải nghiệm người dùng mượt mà với các chức năng hiển thị rõ ràng nước đi, điểm số và trạng thái game.
- Đảm bảo hiệu năng và khả năng tương tác người dùng cao.

1.2.2 Mục tiêu học tập

Thông qua quá trình thực hiện đồ án, xác định được :

- Hiểu và vận dụng lập trình hướng đối tượng (OOP): Thiết kế lớp, kế thừa, đóng gói và đa hình trong dự án thực tế.
- Thực hành mô hình thiết kế phần mềm MVC: Tách biệt rõ ràng giữa phần logic (Model), phần hiển thị (View) và điều khiển (Controller).
- Nâng cao kỹ năng Java Swing: Tạo giao diện đồ họa thân thiện, sự kiện tương tác.
- Hiểu và xây dựng thuật toán trò chơi: Cài đặt thuật toán kiểm tra nước đi, lật quân, tính điểm và cơ chế hoạt động của AI đơn giản.
- Phát triển kỹ năng giải quyết vấn đề thực tế: Thiết kế giải pháp từ đầu, xử lý lỗi và kiểm thử chương trình.

1.3 Yêu cầu chức năng

Bảng 1-1. Bảng yêu cầu chức năng của trò chơi Othello

Mã chức năng	Tên yêu cầu chức năng	Mô tả chi tiết
RF1	Quản lý Game Board	
RF1.1	Vẽ bàn cờ	Vẽ bàn cờ 8x8 rõ ràng, sử dụng đường kẻ để phân chia các ô.
RF1.2	Hiển thị quân cờ	Hiển thị quân cờ đen/trắng tương ứng với trạng thái game.
RF1.3	Đánh dấu nước đi hợp lệ	Đánh dấu các ô là nước đi hợp lệ cho người chơi hiện tại.
RF1.4	Cập nhật bàn cờ	Cập nhật bàn cờ mỗi khi có nước đi mới.
RF2	Xử lý Lượt Chơi	
RF2.1	Chuyển đổi chế độ chơi	Chọn chế độ chơi hai người chơi hoặc giữa người chơi và AI.
RF2.2	Kiểm tra tính hợp lệ	Kiểm tra tính hợp lệ của mỗi nước đi.
RF2.3	Thực hiện lật quân	Thực hiện thao tác lật quân theo đúng quy tắc.
RF3	Chế độ Chơi	
RF3.1	Chế độ 2 người chơi	Chế độ 2 người chơi trên cùng máy (PvP).
RF3.2	Chế độ chơi với AI	Chế độ chơi với máy tính – AI sử dụng thuật toán random.
RF3.3	Chuyển đổi chế độ	Cho phép chuyển đổi giữa các chế độ trong lúc chơi.
RF4	Quản lý Trạng thái Game	
RF4.1	Hiển thị lượt chơi	Hiển thị lượt chơi hiện tại (Black/White).
RF4.2	Phát hiện kết thúc game	Phát hiện khi trò chơi kết thúc.
RF4.3	Thông báo thắng/hòa	Thông báo người chiến thắng hoặc hòa.
RF4.4	Đếm và hiển thị quân	Đếm và hiển thị số quân mỗi bên.
RF5	Điều khiển Game	
RF5.1	Reset trò chơi	Reset trò chơi bằng phím R.
RF5.2	Chuyển đổi AI	Chuyển đổi AI bằng phím A.
RF5.3	Thoát game	Thoát game bằng phím ESC.
RF5.4	Chọn nước đi bằng chuột	Click chuột để chọn nước đi.
RF6	AI System	
RF6.1	AI chọn nước đi	AI chọn ngẫu nhiên một nước đi hợp lệ.
RF6.2	AI thực hiện nước đi	AI thực hiện nước đi và lật quân như người chơi thật.
RF6.3	AI phản hồi nhanh	Đảm bảo AI phản hồi nhanh (dưới 500ms).

1.4 Yêu cầu phi chức năng

1.4.1 Yêu cầu về hiệu năng

- NFR1: Thời gian phản hồi cho mỗi thao tác < 100ms
- NFR2: Game chạy mượt mà với frame rate ổn định
- NFR3: Bộ nhớ sử dụng tối ưu cho ứng dụng desktop

1.4.2 Yêu cầu về giao diện

- NFR4: Giao diện đơn giản, trực quan
- NFR5: Màu sắc tương phản rõ ràng (đen/trắng)
- NFR6: Kích thước cửa sổ cố định 500x600 pixels
- NFR7: Font chữ rõ ràng, kích thước phù hợp

1.4.3 Yêu cầu về tính ổn định

- NFR8: Không crash trong quá trình chơi bình thường
- NFR9: Xử lý lỗi đầu vào từ người dùng
- NFR10: Tương thích với Java JDK 14+

1.4.4 Yêu cầu về khả năng sử dụng

- NFR11: Dễ dàng học cách sử dụng cho người mới
- NFR12: Các phím tắt được ghi chú rõ ràng
- NFR13: Feedback tức thì cho các thao tác của người dùng

1.5 Công cụ sử dụng

1.5.1 Ngôn ngữ lập trình

- Java JDK 14: Ngôn ngữ lập trình chính, đảm bảo tính ổn định và khả năng đa nền tảng
- Java Swing: Thư viện GUI để tạo giao diện game với JFrame và JPanel
- Java AWT: Xử lý đồ họa, vẽ bàn cờ và các sự kiện mouse/keyboard

1.5.2 Môi trường phát triển

- Visual Studio Code (VS Code): Là môi trường phát triển chính được sử dụng để viết mã, biên dịch và debug chương trình. VS Code nhẹ, linh hoạt và có thể mở rộng thông qua các extension.
 - Cài đặt Java Extension Pack từ Marketplace, bao gồm các công cụ hỗ trợ biên dịch, chạy và debug Java.
 - Cấu hình JDK 14 trong biến môi trường JAVA_HOME để đảm bảo VS Code nhận diện đúng phiên bản JDK.
 - Sử dụng các file cấu hình như launch.json và tasks.json để chạy và debug ứng dụng dễ dàng.
- Git: Được sử dụng để quản lý phiên bản mã nguồn, theo dõi lịch sử thay đổi và làm việc nhóm hiệu quả thông qua các nền tảng như GitHub hoặc GitLab.

CHƯƠNG 2. THIẾT KẾ HỆ THỐNG

2.1 Kiến trúc tổng thể

Phần này trình bày cấu trúc tổng quan, định nghĩa các thành phần chính và mô tả mối quan hệ tương tác giữa các chức năng chính



Hình 2-1. Sơ đồ phân cấp chức năng

2.1.1 Giao diện người dùng (UI - User Interface)

Module này chịu trách nhiệm hiển thị trạng thái trò chơi trực quan cho người dùng, bao gồm bàn cờ, vị trí các quân cờ, điểm số, và các thông báo trạng thái. Đồng thời, nó xử lý các tương tác đầu vào từ người dùng (ví dụ: sự kiện click chuột) và chuyển đổi chúng thành các hành động điều khiển cho logic trò chơi.

2.1.2 Quản lý trò chơi (Game logic)

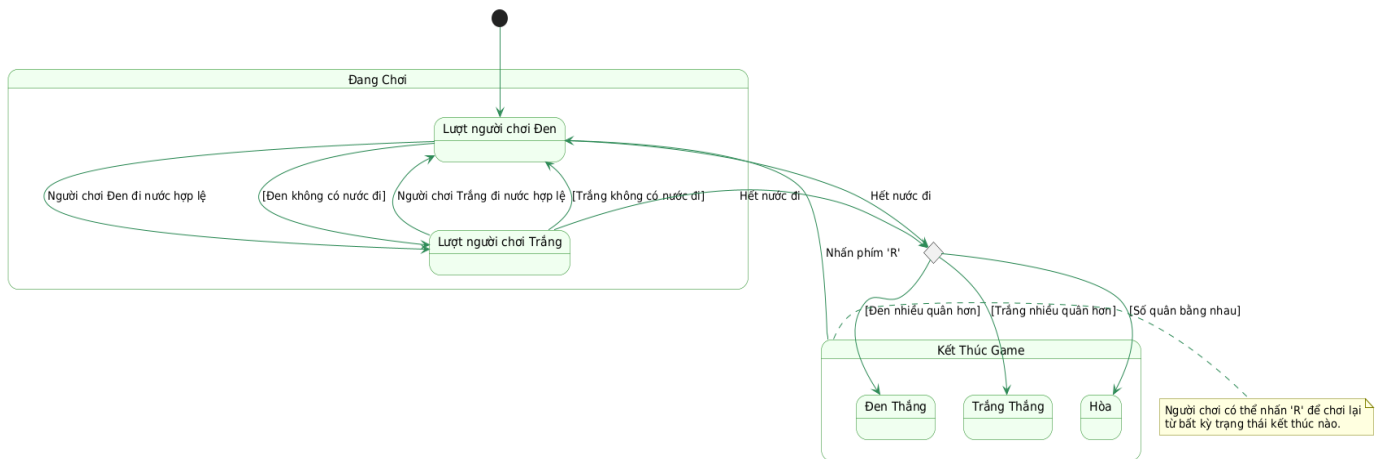
Đây là module cốt lõi chứa các quy tắc và thuật toán của trò chơi Othello. Nó quản lý trạng thái của bàn cờ, xử lý lượt đi của người chơi hoặc AI, kiểm tra tính hợp lệ của các nước đi, thực hiện lật quân cờ, và xác định điều kiện kết thúc trò chơi. Module này cũng cung cấp chức năng khởi động lại trò chơi.

2.1.3 Trí tuệ nhân tạo đơn giản (Simple AI)

Module này cung cấp khả năng chơi tự động cho đối thủ máy tính. Chức năng chính của nó là phân tích trạng thái bàn cờ hiện tại và xác định một nước đi hợp lệ theo thuật toán được cài đặt (ví dụ: chọn ngẫu nhiên một nước đi hợp lệ) để đối phó với người chơi.

2.2 Sơ đồ trạng thái (State diagram)

Phần này mô tả các trạng thái chính của trò chơi và các chuyển đổi giữa chúng.



Hình 2-2. Sơ đồ trạng thái của trò chơi Othello

Trạng thái bắt đầu (Initial State): Điểm khởi đầu của hệ thống.

Trạng thái Đang Chơi (Playing State): Trạng thái chính khi game đang diễn ra, bao gồm các trạng thái con:

- Lượt người chơi Đen: Chuyển sang Lượt người chơi Trắng nếu Đen đi hợp lệ hoặc Đen không có nước đi.
- Lượt người chơi Trắng: Chuyển sang Lượt người chơi Đen nếu Trắng đi hợp lệ hoặc Trắng không có nước đi.
- Chuyển ra: Sang Kết Thúc Game khi "Hết nước đi" (cả hai bên không còn nước đi hợp lệ).

Trạng thái Kết Thúc Game (Game Over State): Giai đoạn cuối của trò chơi, bao gồm các trạng thái con:

- Đen Thắng: Khi Đen nhiều quân hơn.
- Trắng Thắng: Khi Trắng nhiều quân hơn.
- Hòa: Khi số quân bằng nhau.
- Chuyển tiếp: Từ bất kỳ trạng thái kết thúc nào, nhấn "Nhấn phím 'R'" sẽ khởi động lại game, quay về trạng thái Đang Chơi.

2.3 Sơ đồ hoạt động (FlowChart)

Sơ đồ hoạt động minh họa luồng điều khiển nghiệp vụ của trò chơi Othello, chi tiết hóa quy trình tương tác giữa người chơi và hệ thống qua các chế độ chơi khác nhau :

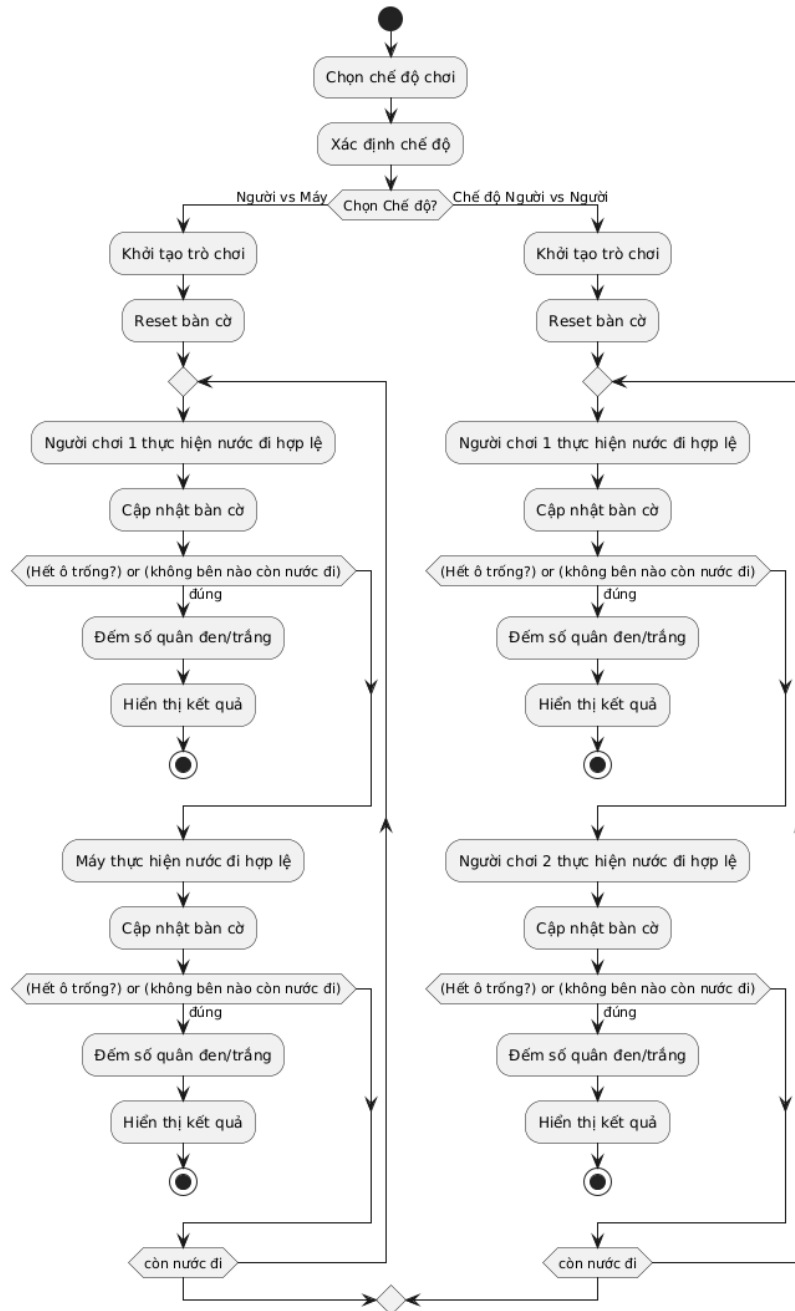
Khởi tạo trò chơi: Quá trình bắt đầu với việc người dùng Chọn chế độ chơi (Người vs Máy hoặc Người vs Người), sau đó hệ thống sẽ Khởi tạo trò chơi / Reset bàn cờ tương ứng.

Vòng lặp chơi chính: Sau khi khởi tạo, trò chơi đi vào một vòng lặp lặp lại cho mỗi lượt chơi:

- Lượt của Người chơi 1: Người chơi thực hiện nước đi hợp lệ.
- Cập nhật trạng thái: Hệ thống Cập nhật bàn cờ, Kiểm tra kết thúc (hết ô trống hoặc không bên nào còn nước đi), Đếm số quân đen/trắng, và Hiển thị kết quả.

- Rẽ nhánh chế độ chơi:
 - Người vs Máy: Nếu trò chơi chưa kết thúc, Máy thực hiện nước đi hợp lệ. Sau đó, lại lặp lại quy trình cập nhật và kiểm tra trạng thái.
 - Người vs Người: Nếu trò chơi chưa kết thúc, Người chơi 2 thực hiện nước đi hợp lệ. Sau đó, lại lặp lại quy trình cập nhật và kiểm tra trạng thái.
- Cả hai chế độ đều quay lại lượt của Người chơi 1 nếu trò chơi còn nước đi.

Kết thúc trò chơi: Luồng hoạt động kết thúc khi điều kiện Hết ô trống hoặc Không bên nào còn nước đi được đáp ứng, dẫn đến việc hiển thị kết quả cuối cùng.



Hình 2-3. Sơ đồ hoạt động của trò chơi Othello

- +handleInput(int keyCode): Xử lý các sự kiện đầu vào từ bàn phím.
- +playTurn(Position gridPosition): Thực hiện một lượt chơi dựa trên vị trí được chọn.
- +restart(): Khởi tạo lại trò chơi về trạng thái ban đầu.

2.4.2 GridGame

- Mô tả: Lớp cốt lõi chứa toàn bộ logic và trạng thái của bàn cờ Othello. Đây là nơi các quy tắc trò chơi được thực thi.
- Thuộc tính quan trọng:
 - -grid: GridCell[][]: Mảng 2 chiều đại diện cho 64 ô trên bàn cờ Othello (8x8).
- Phương thức quan trọng:
 - +playMove(Position movePosition, int player): Thực hiện một nước đi của người chơi xác định tại vị trí đã cho.
 - +isValidMove(Position position, boolean isPlayerTurn): Kiểm tra xem một nước đi có hợp lệ tại vị trí đã cho hay không.
 - +flipPieces(Position movePosition, int player): Thực hiện lật các quân cờ đối thủ sau một nước đi hợp lệ.
 - +updateValidMoves(int playerID): Cập nhật danh sách các nước đi hợp lệ cho người chơi hiện tại.
 - +getWinner(boolean stillValidMoves): Xác định người chiến thắng hoặc trạng thái hòa khi trò chơi kết thúc.

2.4.3 GridCell

- Mô tả: Đại diện cho một ô đơn lẻ trên bàn cờ Othello.
- Thuộc tính quan trọng:
 - -cellState: int: Trạng thái của ô cờ (ví dụ: 0-trống, 1-đen, 2-trắng).
 - -highlight: boolean: Cờ hiệu chỉ thị ô có đang được làm nổi bật hay không (ví dụ: cho các nước đi hợp lệ).
- Phương thức quan trọng:
 - +paint(Graphics g): Vẽ trạng thái của ô cờ (quân đen, quân trắng hoặc ô trống).
 - +setCellState(int newState): Cập nhật trạng thái của ô cờ.

2.4.4 Position

- Mô tả: Lớp tiện ích đơn giản để biểu diễn tọa độ (x, y) trên lưới bàn cờ.
- Thuộc tính: +x: int, +y: int.
- Phương thức: Constructor +Position(int x, int y).

2.4.5 Rectangle

- Mô tả: Lớp biểu diễn một hình chữ nhật, có thể được sử dụng để định nghĩa kích thước và vị trí của các ô cờ hoặc các thành phần đồ họa khác trên giao diện.
- Thuộc tính: #position: Position, #width: int, #height: int.
- Mỗi quan hệ: GridCell "kế thừa" hoặc "có một" Rectangle (mỗi quan hệ composition hoặc association) để định nghĩa khu vực vẽ của nó.

2.4.6 SimpleAI

- Mô tả: Lớp cài đặt trí tuệ nhân tạo ở mức cơ bản, chịu trách nhiệm chọn nước đi cho đối thủ máy tính.
- Thuộc tính quan đó:
 - -gameGrid: GridGame: Tham chiếu đến đối tượng bàn cờ để AI có thể đọc trạng thái hiện tại và tìm nước đi.
- Phương thức quan trọng:
 - +chooseMove(): Phương thức chính để AI xác định và trả về một Position của nước đi mà nó muốn thực hiện (ví dụ: chọn ngẫu nhiên một nước đi hợp lệ).

2.4.7 Giao diện và Lớp cơ sở (Interfaces and Base Classes):

- MouseListener, KeyListener: Các giao diện chuẩn của Java AWT/Swing để xử lý các sự kiện chuột và bàn phím tương ứng.
- JPanel: Lớp cơ sở của Java Swing để tạo các thành phần giao diện đồ họa. GamePanel kế thừa từ lớp này.
- Game: (Được minh họa như một Interface) Có thể định nghĩa một hợp đồng chung cho các trò chơi, bao gồm các phương thức như main() để khởi động ứng dụng và keyPressed() để xử lý đầu vào cấp cao.

CHƯƠNG 3. TRIỂN KHAI CÀI ĐẶT TRÒ CHƠI

3.1 Các chức năng đã triển khai

Trong phần này sẽ trình bày chi tiết về việc triển khai trò chơi Othello, bao gồm các chức năng cốt lõi từ giao diện người dùng đến logic trò chơi và AI đơn giản:

Bảng 3-1. Bảng chức năng đã triển khai

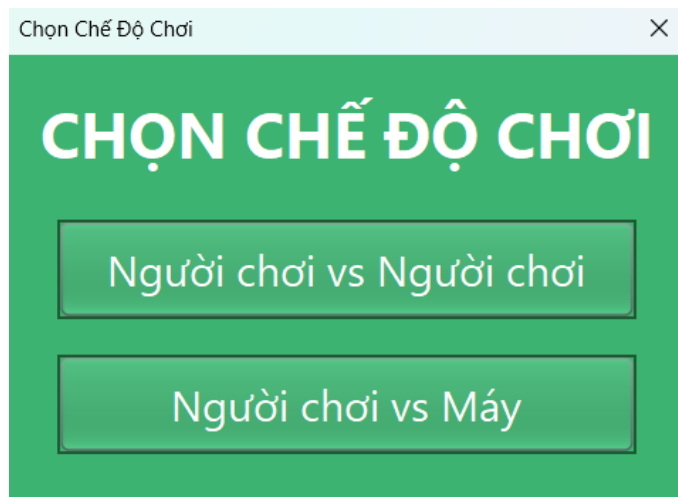
Chức năng	Mô tả chi tiết
Khởi tạo và Quản lý Trò chơi	<ul style="list-style-type: none">- Tạo cửa sổ ứng dụng Othello- Thiết lập giao diện người dùng theo phong cách Nimbus- Xử lý các sự kiện bàn phím (thoát game: ESC, khởi động lại: R, chọn chế độ: A)- Hộp thoại lựa chọn chế độ chơi (Người chơi vs Người chơi hoặc Người chơi vs Máy)
Lưới Trò chơi và Logic	<ul style="list-style-type: none">- Tạo lưới Othello với kích thước và vị trí xác định.- Lưu trữ trạng thái từng ô (trống, quân đen, quân trắng)- Chuyển đổi vị trí nhấp chuột thành tọa độ lưới.- Xác định và cập nhật nước đi hợp lệ, bao gồm tô sáng ô hợp lệ- Thực hiện nước đi: đặt quân cờ và lật quân đối thủ- Tính toán số lượng quân cờ mỗi người chơi- Xác định kết quả trò chơi (Thắng/Thua/Hòa)
Trạng thái Trò chơi và Hiển thị	<ul style="list-style-type: none">- Quản lý các trạng thái game (Lượt Đen, Lượt Trắng)- Hiển thị thông báo trạng thái game hiện tại- Hiển thị số lượng quân cờ của Đen và Trắng- Vẽ lưới trò chơi và các quân cờ- Tô sáng các ô hợp lệ
Đối tượng và Tiện ích Chung	<ul style="list-style-type: none">- Đại diện cho cặp tọa độ (x, y) và các phép toán cơ bản- Đại diện cho hình chữ nhật (vị trí, chiều rộng, chiều cao) với các phương thức kiểm tra
AI đơn giản	<ul style="list-style-type: none">- AI đơn giản chọn ngẫu nhiên một nước đi hợp lệ

Các chức năng đã triển khai không chỉ đáp ứng đầy đủ các yêu cầu cơ bản của trò chơi Othello, từ giao diện người dùng, xử lý logic trò chơi đến hỗ trợ AI đơn giản, mà còn đảm bảo khả năng tương tác và trải nghiệm người dùng mượt mà.

3.2 Giao diện trò chơi

3.2.1 Giao diện chọn chế độ chơi

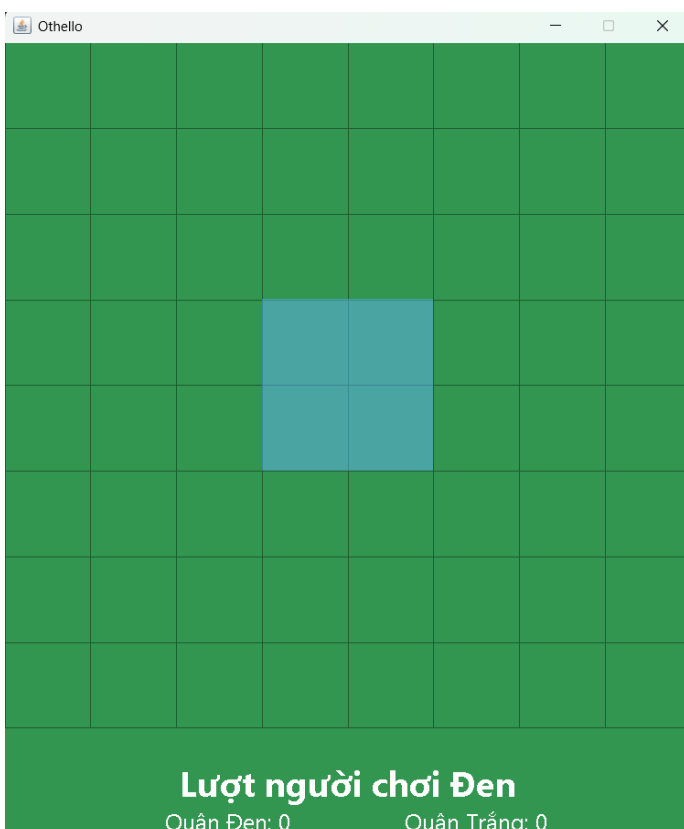
Giao diện chọn chế độ chơi cho phép người dùng lựa chọn giữa hai hình thức: Người chơi vs Người chơi hoặc Người chơi vs Máy. Giao diện được thiết kế đơn giản, trực quan với hai nút chức năng rõ ràng, giúp người dùng dễ dàng thao tác trước khi bắt đầu trò chơi. Đây là bước đầu tiên trong quá trình thiết lập một ván Othello mới.



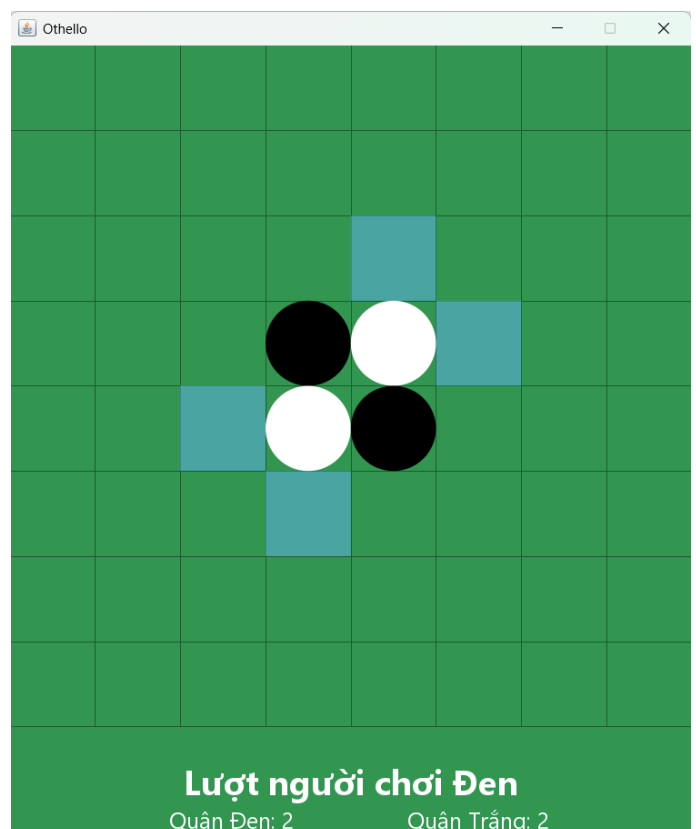
Hình 3-1. Giao diện chọn chế độ chơi

3.2.2 Giao diện bàn cờ

Sau khi chọn chế độ chơi, ở Hình 3-2 bạn có thể thấy bàn cờ ban đầu với bốn ô giữa được tô màu xanh nhạt, cho biết đây là những vị trí hợp lệ duy nhất để đặt những quân cờ đầu tiên. Tại thời điểm này, cả quân Đen và quân Trắng đều chưa có quân nào trên bàn.



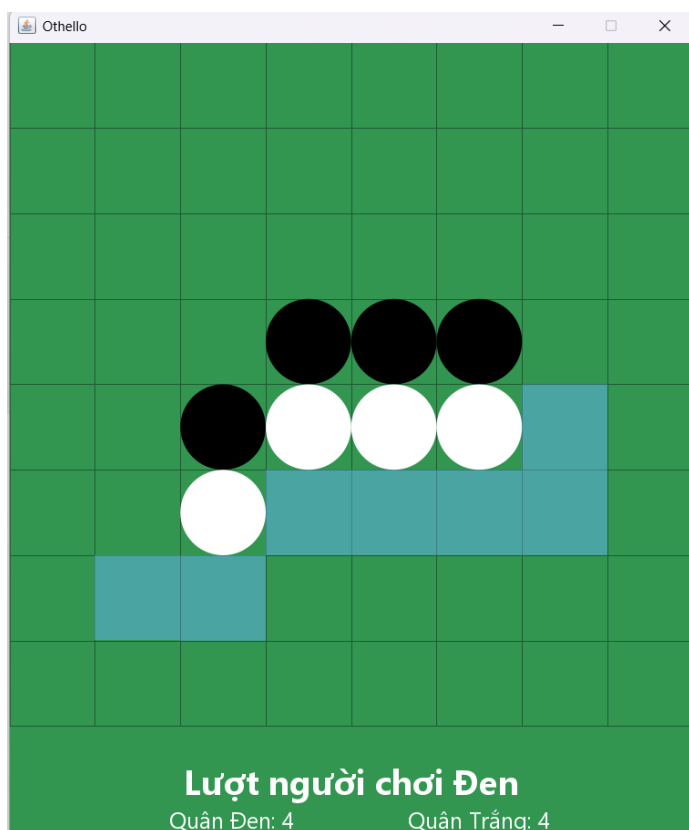
Hình 3-2. Giao diện bàn cờ 1



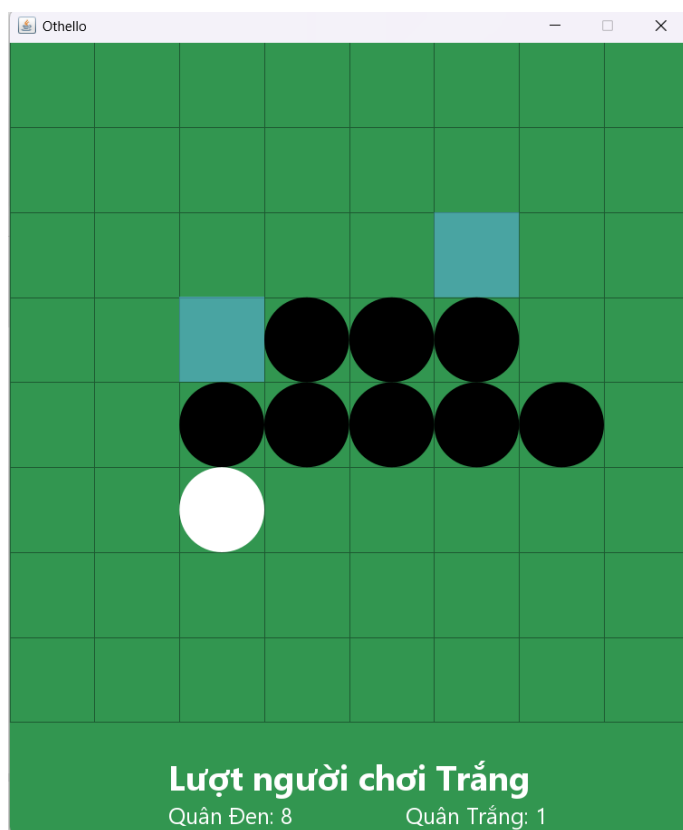
Hình 3-3. Giao diện bàn cờ 2

Sang Hình 3-3, sau khi bốn quân đầu tiên đã được đặt, bạn có thể thấy các ô hợp lệ để đánh tiếp theo (cũng được tô màu xanh nhạt) xuất hiện bên ngoài vùng trung tâm ban đầu. Lúc này, đến lượt người chơi Đen, và nếu quân Đen được đặt

tại bất kỳ vị trí tô sáng nào, nó sẽ lật một quân Trắng nằm giữa hai quân Đen theo quy tắc của trò chơi Othello.



Hình 3-4. Giao diện bàn cờ 3



Hình 3-5. Giao diện bàn cờ 4

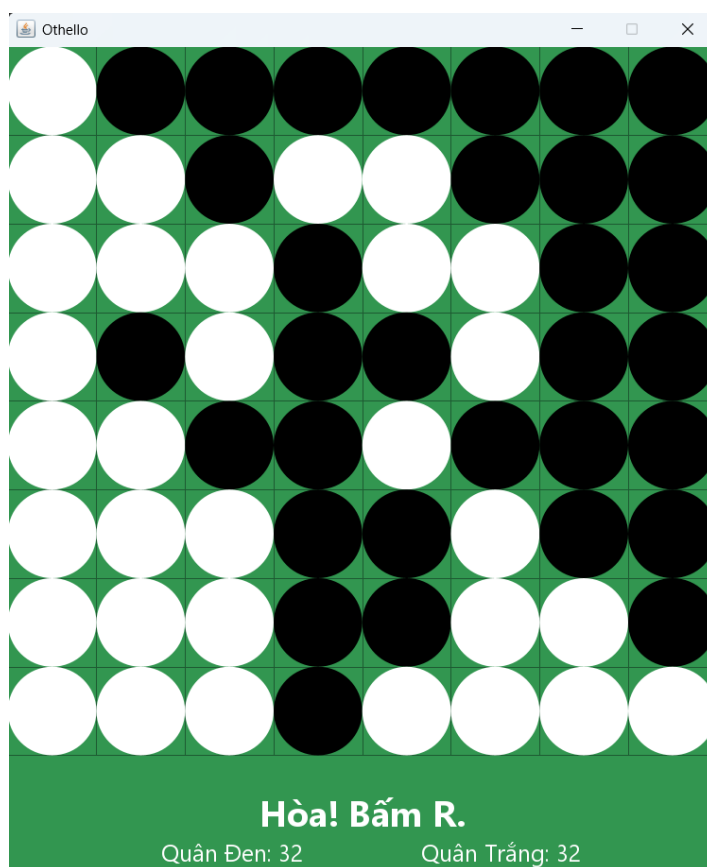
Hình 3-4 minh họa lượt chơi của người chơi Đen khi có bốn quân Đen và bốn quân Trắng đang hiện diện trên bàn. Các ô màu xanh nhạt thể hiện những vị trí hợp lệ để đặt quân Đen. Nếu người chơi Đen đặt quân vào ô ngoài cùng bên phải hàng ngang, toàn bộ các quân Trắng nằm giữa hai quân Đen theo hàng ngang sẽ bị lật thành màu Đen.

Hình 3-5 cho thấy kết quả sau khi nước đi đó được thực hiện: tất cả quân Trắng nằm trên đường giữa hai quân Đen đã bị lật và chuyển thành quân Đen. Lúc này, số lượng quân Đen đã tăng lên 8, trong khi quân Trắng chỉ còn lại 1. Lượt chơi tiếp theo thuộc về người chơi Trắng, và các ô hợp lệ mới được đánh dấu bằng màu xanh nhạt.

Trò chơi sẽ tiếp tục luân phiên lượt giữa hai người chơi cho đến khi không còn nước đi hợp lệ nào cho cả hai bên. Điều này có thể xảy ra do không còn ô hợp lệ nào để đi hoặc toàn bộ bàn cờ đã được lấp đầy. Bàn cờ không nhất thiết phải đầy hoàn toàn để trò chơi kết thúc.

Hình 3-6 thể hiện kết quả cuối cùng của một ván cờ Othello khi trò chơi kết thúc với tỷ số hòa. Bàn cờ đã được lấp đầy hoàn toàn với tổng cộng 64 quân cờ, trong đó người chơi Đen và người chơi Trắng đều chiếm 32 quân.

Dòng thông báo "*Hòa! Bấm R.*" hiển thị rõ ràng kết quả trận đấu. Điều này minh họa cho một trong ba trường hợp kết thúc có thể xảy ra trong trò chơi Othello: người chơi Đen thắng, người chơi Trắng thắng, hoặc hòa – tùy thuộc vào số lượng quân mỗi bên kiểm soát được khi không còn nước đi hợp lệ nào nữa.



Hình 3-6. Giao diện bàn cờ 5

3.3 Code hoàn chỉnh

Toàn bộ mã nguồn của trò chơi Othello đã được xây dựng đầy đủ, bao gồm giao diện người dùng, xử lý luật chơi, cập nhật lượt đánh và kiểm tra điều kiện kết thúc trận đấu. Chương trình được cài đặt trên nền tảng Java Swing.

Chi tiết mã nguồn đã được đăng tải tại kho lưu trữ GitHub theo liên kết dưới đây:

<https://github.com/VuxVux22/Othello>

Người đọc có thể truy cập vào đường dẫn trên để xem toàn bộ mã, cách cài đặt và hướng dẫn sử dụng trò chơi.

CHƯƠNG 4. KIỂM THỬ

4.1 Kiểm thử chức năng

Phần kiểm thử chức năng được thực hiện nhằm đánh giá mức độ đáp ứng của hệ thống đối với các yêu cầu chức năng (RF) đã được xác định trước đó. Quá trình này giúp xác minh rằng các chức năng chính của hệ thống hoạt động đúng như thiết kế và không phát sinh lỗi trong quá trình vận hành. Kết quả kiểm thử được tổng hợp trong bảng dưới đây để làm cơ sở đánh giá chất lượng phần mềm.

Bảng 4-1. Bảng kiểm thử chức năng

Chức năng kiểm thử	Mô tả kiểm thử	Kết quả mong đợi	Trạng thái
RF1: Quản lý Game Board			
Hiển thị bàn cờ	Kiểm tra bàn cờ 8x8 với lưới rõ ràng khi khởi động game.	Bàn cờ 8x8 với các đường lưới rõ ràng được hiển thị.	ĐẠT
Hiển thị quân cờ khởi tạo	Kiểm tra vị trí và màu sắc 4 quân cờ ban đầu.	4 quân cờ được đặt ở giữa bàn cờ theo pattern criss-cross (đen-trắng-trắng-đen).	ĐẠT
Highlight ô hợp lệ	Kiểm tra các ô hợp lệ được tô sáng.	Các ô có thể đặt quân (valid moves) được tô sáng (màu xanh nhạt).	ĐẠT
Cập nhật trạng thái bàn cờ	Kiểm tra bàn cờ cập nhật sau mỗi nước đi.	Trạng thái bàn cờ được cập nhật theo thời gian thực (quân cờ được đặt, các quân bị lật).	ĐẠT
RF2: Xử lý Lượt Chơi			
Quản lý lượt chơi	Kiểm tra chuyển đổi lượt giữa hai người chơi.	Lượt chơi tự động chuyển đổi giữa Người chơi Đen và Người chơi Trắng. Người chơi quân đen đi trước.	ĐẠT
Validate nước đi hợp lệ	Kiểm tra hệ thống từ chối nước đi không hợp lệ.	Quân cờ không được đặt, không có quân nào bị lật khi cố gắng đặt quân vào ô không hợp lệ.	ĐẠT
Thực hiện lật quân	Kiểm tra lật quân đối thủ sau nước đi hợp lệ.	Tất cả quân của đối thủ bị kẹp sẽ được lật thành màu của người chơi hiện tại.	ĐẠT
RF3: Chế độ Chơi			
Chế độ Player vs Player (PvP)	Kiểm tra khả năng chơi giữa hai người.	Game cho phép hai người chơi luân phiên điều khiển.	ĐẠT

Chế độ Player vs AI (PvAI)	Kiểm tra khả năng chơi với máy tính (AI random).	Game cho phép người chơi đấu với AI đơn giản. AI thực hiện nước đi ngẫu nhiên nhưng hợp lệ.	ĐẠT
Chuyển đổi chế độ chơi	Kiểm tra khả năng chuyển đổi chế độ chơi trong game.	Hộp thoại chọn chế độ chơi xuất hiện khi nhấn phím 'A', cho phép chuyển đổi giữa PvP và PvAI.	ĐẠT
RF4: Quản lý Trạng thái Game			
Hiển thị thông báo lượt chơi	Kiểm tra thông báo lượt chơi hiện tại.	Dòng thông báo hiển thị lượt của người chơi hiện tại (ví dụ: "Lượt Đen", "Lượt Trắng").	ĐẠT
Phát hiện và thông báo kết thúc game	Kiểm tra thông báo kết thúc game.	Game hiển thị thông báo "Hòa! Bấm R.", "Đen Thắng! Bấm R." hoặc "Trắng Thắng! Bấm R." khi game kết thúc.	ĐẠT
Xác định và hiển thị người thắng/hòa	Kiểm tra kết quả thắng/thua/hòa.	Kết quả đúng (Đen thắng, Trắng thắng, hoặc Hòa) dựa trên số quân cuối cùng.	ĐẠT
Đếm và hiển thị số quân	Kiểm tra số lượng quân của mỗi bên.	Số lượng quân Đen và quân Trắng được hiển thị chính xác và cập nhật theo thời gian thực.	ĐẠT
RF5: Điều khiển Game			
Reset game	Kiểm tra chức năng reset game bằng phím 'R'.	Game khởi động lại về trạng thái ban đầu, bàn cờ được reset.	ĐẠT
Thoát game	Kiểm tra chức năng thoát game bằng phím 'ESC'.	Ứng dụng game đóng lại.	ĐẠT
Đặt quân bằng chuột	Kiểm tra khả năng đặt quân bằng click chuột.	Quân cờ được đặt tại vị trí click và các quân đối thủ bị lật nếu có.	ĐẠT
RF6: AI System			
AI chọn nước đi hợp lệ	Kiểm tra AI luôn chọn nước đi hợp lệ.	AI luôn thực hiện một nước đi hợp lệ (đặt quân vào ô tô sáng).	ĐẠT
AI thực hiện nước đi tự động	Kiểm tra AI tự động thực hiện nước đi.	AI tự động đặt quân cờ mà không cần tương tác từ người dùng.	ĐẠT
Thời gian phản hồi của AI	Kiểm tra thời gian phản hồi của AI.	Thời gian phản hồi của AI là hợp lý (dưới 100ms với AI random).	ĐẠT

4.2 Kiểm thử phi chức năng

Kiểm thử phi chức năng nhằm đánh giá các khía cạnh không liên quan trực tiếp đến chức năng của hệ thống như hiệu năng, tính bảo mật, khả năng sử dụng và khả năng tương thích. Mục tiêu là đảm bảo hệ thống hoạt động ổn định, hiệu quả trong nhiều điều kiện khác nhau và mang lại trải nghiệm người dùng tốt. Các nội dung kiểm thử phi chức năng được trình bày cụ thể trong bảng dưới đây.

Bảng 4-2. Bảng kiểm thử phi chức năng

Chức năng kiểm thử	Mô tả kiểm thử	Kết quả mong đợi	Trạng thái
Yêu cầu về hiệu năng			
Thời gian phản hồi thao tác	Đo thời gian phản hồi khi click chuột, nhấn phím.	Thời gian phản hồi cho mỗi thao tác < 100ms.	ĐẠT
Frame rate ổn định	Quan sát độ mượt mà của game.	Game chạy mượt mà, không giật lag, frame rate ổn định.	ĐẠT
Tối ưu bộ nhớ	Kiểm tra mức độ sử dụng bộ nhớ của ứng dụng.	Bộ nhớ sử dụng tối ưu cho ứng dụng desktop, không rò rỉ bộ nhớ đáng kể.	ĐẠT
Yêu cầu về giao diện			
Giao diện trực quan	Đánh giá tính đơn giản và trực quan của giao diện.	Giao diện đơn giản, trực quan, dễ hiểu.	ĐẠT
Màu sắc tương phản	Kiểm tra độ tương phản giữa các thành phần.	Màu sắc tương phản rõ ràng (đen/trắng).	ĐẠT
Kích thước cửa sổ cố định	Kiểm tra kích thước cửa sổ game.	Kích thước cửa sổ game cố định 500x600 pixels.	ĐẠT
Font chữ rõ ràng	Kiểm tra font chữ và kích thước.	Font chữ rõ ràng, kích thước phù hợp, dễ đọc.	ĐẠT
Yêu cầu về tính ổn định			
Không crash	Kiểm tra ứng dụng không bị crash.	Ứng dụng không bị crash trong quá trình chơi bình thường.	ĐẠT
Xử lý lỗi đầu vào	Kiểm tra xử lý đầu vào không hợp lệ (nếu có).	Ứng dụng không bị lỗi, không phản ứng hoặc hiển thị thông báo lỗi phù hợp (nếu có).	ĐẠT
Tương thích JDK 14+	Kiểm tra khả năng chạy trên JDK 14+.	Ứng dụng hoạt động bình thường trên các phiên bản Java JDK 14 trở lên.	ĐẠT
Yêu cầu về khả năng sử dụng			
Dễ học	Đánh giá mức độ dễ dàng cho người mới.	Người chơi mới dễ dàng học cách sử dụng.	ĐẠT
Ghi chú phím tắt	Kiểm tra các phím tắt được ghi chú rõ ràng.	Các phím tắt (R, A, ESC) được ghi chú rõ ràng trong tài liệu hoặc trong giao diện (nếu có).	ĐẠT
Phản hồi tức thì	Kiểm tra phản hồi của hệ thống.	Có phản hồi tức thì cho các thao tác của người dùng.	ĐẠT

CHƯƠNG 5. KẾT LUẬN

5.1 Những gì đã học được

Đồ án "Thiết kế và Xây dựng Trò chơi Othello bằng ngôn ngữ lập trình Java" đã hoàn thành các mục tiêu đề ra, mang lại một ứng dụng trò chơi Othello hoạt động ổn định và đáp ứng các yêu cầu về chức năng cũng như giao diện người dùng.

Thông qua quá trình thực hiện đồ án, em đã đạt được những kiến thức và kỹ năng quan trọng sau:

- Áp dụng Nguyên lý Thiết kế Hướng đối tượng (OOP): Em đã thực hành triển khai các khái niệm OOP như kế thừa, đóng gói, đa hình trong việc xây dựng các lớp như GamePanel, GridGame, GridCell, Position, Rectangle, và SimpleAI để quản lý các thành phần của trò chơi một cách có tổ chức và dễ mở rộng.
- Thực hành Mô hình Thiết kế MVC (Model-View-Controller): Đồ án đã giúp em hiểu và áp dụng mô hình MVC, tách biệt rõ ràng phần logic xử lý (Model - GridGame), giao diện người dùng (View - các phương thức paint trong GamePanel và GridCell), và bộ điều khiển tương tác (Controller - GamePanel xử lý sự kiện MouseListener, KeyListener).
- Phát triển Kỹ năng Xây dựng Giao diện Người dùng với Java Swing: Em đã thành thạo việc sử dụng thư viện Java Swing để tạo cửa sổ ứng dụng (JFrame, JPanel), vẽ các thành phần đồ họa (bàn cờ, quân cờ, highlight ô hợp lệ) và xử lý các sự kiện từ chuột/bàn phím, tạo ra một giao diện trực quan và dễ tương tác.
- Cài đặt Thuật toán Game Logic và AI cơ bản: Em đã tự tay cài đặt các thuật toán cốt lõi của trò chơi Othello như kiểm tra nước đi hợp lệ (isValidMove), lật quân cờ (flipPieces), cập nhật trạng thái bàn cờ, và xác định điều kiện kết thúc game (getWinner). Ngoài ra, em cũng đã xây dựng được một AI cơ bản với khả năng chọn nước đi ngẫu nhiên.

5.2 Định hướng phát triển

Trong tương lai, đồ án có thể được phát triển và mở rộng theo các hướng sau để nâng cao trải nghiệm người dùng và tính phức tạp của trò chơi:

- Nâng cấp Trí tuệ Nhân tạo (AI): Thay vì chỉ chọn ngẫu nhiên, AI có thể được cải thiện bằng cách triển khai các thuật toán tìm kiếm có cây như Minimax hoặc Alpha-Beta Pruning. Điều này sẽ giúp AI có khả năng đánh giá nước đi tốt hơn và đưa ra chiến lược hiệu quả hơn, tạo ra một thử thách thực sự cho người chơi.
- Đa dạng hóa Chế độ Chơi: Bổ sung thêm các chế độ chơi khác như chơi trực tuyến (multiplayer online), hoặc tùy chỉnh độ khó của AI (dễ, trung bình, khó) để người chơi có nhiều lựa chọn hơn.

- Cải thiện Giao diện và Trải nghiệm Người dùng (UX): Nâng cấp đồ họa, thêm hiệu ứng âm thanh, hoạt ảnh khi lật quân, và các tùy chọn cá nhân hóa giao diện (ví dụ: thay đổi màu sắc bàn cờ, kiểu quân cờ) để tăng tính hấp dẫn và mượt mà cho trò chơi.
- Triển khai Chức năng Lưu/Tải Trò chơi: Cho phép người chơi lưu lại trạng thái hiện tại của ván đấu và tải lại sau để tiếp tục chơi, tăng tính tiện lợi.
- Tối ưu hóa Hiệu năng: Với các thuật toán AI phức tạp hơn, cần có sự tối ưu hóa về mặt hiệu năng để đảm bảo game vẫn chạy mượt mà và thời gian phản hồi hợp lý.

TÀI LIỆU THAM KHẢO

- [1] Kevin Lu, *The Game Theory of Reversi*, Duke University, Department of Mathematics, 2014.
- [2] John Zukowski, *The Definitive Guide to Java Swing*, Apress, 2005.
- [3] Colgate University, *Java JDK and VSCode Installation Guide*, 2022.