# **I.MỤC LỤC**

[**I.MỤC LỤC** 1](#_Toc199430046)

[**II.LỜI CẢM ƠN** 2](#_Toc199430047)

[**II.NỘI DUNG BÀI TẬP LỚN** 3](#_Toc199430048)

[**Chương 1.Giới thiệu đầu bài** 3](#_Toc199430049)

[***1.1.Thông tin đề tài*** 3](#_Toc199430050)

[**1.2.Phân tích bài toán** 3](#_Toc199430051)

[***1.3.Yêu cầu chức năng*** 4](#_Toc199430052)

[**Chương 2.Cơ sở lý thuyết** 5](#_Toc199430053)

[***2.1.Tkinter – Thư viện GUI cơ bản của Python*** 5](#_Toc199430054)

[***2.2.Thuật toán Minimax*** 5](#_Toc199430055)

[**2.3.Cắt tỉa Alpha-Beta** 6](#_Toc199430056)

[**2.4.Lập trình hướng đối tượng (OOP) trong Python** 7](#_Toc199430057)

[**2.5.Design Pattern MVC cho game** 7](#_Toc199430058)

[**Chương 3.Thiết kế và xây dựng chương trình** 9](#_Toc199430059)

[**3.1.Kiến trúc chương trình** 9](#_Toc199430060)

[**3.2.Các chức năng chính** 9](#_Toc199430061)

[**3.3.Sơ đồ UML** 9](#_Toc199430062)

[**3.4.Thiết kế giao diện người dùng** 9](#_Toc199430063)

[**Chương 4.Thực nghiệm và kết luận** 10](#_Toc199430064)

[***4.1.Kết quả thực nghiệm*** 10](#_Toc199430065)

[***4.2.Đánh giá và cải tiến*** 10](#_Toc199430066)

[**III.TÀI LIỆU THAM KHẢO** 11](#_Toc199430067)

# **II.LỜI CẢM ƠN**

Trước hết, em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Tuấn Linh, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện bài tập lớn này. Những đóng góp quý báu và sự chỉ dẫn tận tâm của thầy đã giúp em hoàn thành đề tài một cách hiệu quả và có định hướng rõ ràng

Em cũng xin cảm ơn các thầy cô trong Khoa Điện Tử, Bộ môn Công Nghệ Thông Tin đã tạo điều kiện thuận lợi để em được học tập, thực hành và phát triển kỹ năng lập trình qua đề tài thực tế

Cuối cùng, em xin cảm ơn gia đình và bạn bè đã luôn cổ vũ, động viên tinh thần em trong suốt quá trình học tập và hoàn thành đồ án

Thái Nguyên, ngày….tháng…..năm 2025

**SINH VIÊN**

*(Ký ghi rõ họ tên)*

# **II.NỘI DUNG BÀI TẬP LỚN**

## **Chương 1.Giới thiệu đầu bài**

## ***1.1.Thông tin đề tài***

- Tên đề tài: Phát triển trò chơi “Four in a Row” bằng ngôn ngữ Python

- Mục tiêu: Xây dựng một trò chơi đơn giản theo kiểu “Connect Four” cho phép người chơi đấu với nhau hoặc đấu với máy

## **1.2.Phân tích bài toán**

"Nối 4" (hay Four in a Row hoặc Connect Four) là một trò chơi chiến lược dành cho hai người chơi trên một lưới hình chữ nhật gồm 6 hàng và 7 cột. Người chơi thay phiên nhau thả quân cờ vào một cột bất kỳ trên bàn cờ. Quân cờ sẽ rơi xuống vị trí thấp nhất có thể trong cột đó

Mục tiêu của trò chơi là xếp được 4 quân cờ liên tiếp theo hàng ngang, hàng dọc hoặc đường chéo, trước đối thủ để giành chiến thắng

-Các bước chơi:

+Bắt đầu trò chơi: Trò chơi khởi động với bàn cờ trống một ma trận gồm 6x7 ô. Hai người chơi hoặc người chơi đấu với máy luân phiên thực hiện nước đi. Người chơi đầu tiên được chọn ngẫu nhiên hoặc theo quy tắc đã định trước

+Lượt chơi của người chơi: Người chơi chọn một cột trên bàn cờ để thả quân cờ của mình. Nếu cột đã đầy, người chơi phải chọn một cột khác. Quân cờ sẽ tự động rơi xuống vị trí thấp nhất có thể trong cột đó

+Kiểm tra thắng-thua: Sau mỗi nước đi, chương trình kiểm tra xem có chuỗi 4 quân liên tiếp theo hàng ngang, hàng dọc hoặc đường chéo hay không. Nếu có, trò chơi kết thúc và người chơi vừa thực hiện nước đi sẽ thắng. Nếu bàn cờ đã đầy mà không có người chiến thắng, trò chơi kết thúc với kết quả hòa

+Chế độ chơi với A.I: Khi đấu với máy, chương trình sẽ sử dụng thuật toán Minimax có cắt tỉa Alpha-Beta để tính toán nước đi tối ưu. AI sẽ phân tích các nước đi có thể thực hiện và chọn nước đi có lợi thế nhất để cố gắng giành chiến thắng hoặc cản trở đối thủ

-Các lưu ý: Không thể thả quân cờ vào cột đã đầy. Người chơi không thể rút lại nước đi đã thực hiện. Người chơi không thể rút lại nước đi đã thực hiện. Chỉ người chơi vừa đi mới có thể thắng ngay sau lượt chơi của họ. Nếu tất cả các ô trên bàn cờ đều bị lấp đầy mà chưa có ai thắng, trò chơi kết thúc với kết quả hòa

## ***1.3.Yêu cầu chức năng***

- Giao diện người dùng đơn giản, dễ sử dụng

- Cho phép chơi hai người hoặc một người với máy

- Hiển thị bàn cờ dạng lưới, kiểm tra thắng thua theo hàng, cột, đường chéo

- Áp dụng thuật toán AI cơ bản (minimax có cắt tỉa alpha-beta)

## **Chương 2.Cơ sở lý thuyết**

## ***2.1.Tkinter – Thư viện GUI cơ bản của Python***

Tkinter là thư viện giao diện người dùng (GUI) mặc định của Python, giúp tạo ứng dụng với giao diện đồ họa một cách dễ dàng. Đây là một wrapper của thư viện Tk, cho phép xây dựng các thành phần như cửa sổ, nút bấm, ô nhập văn bản, và các hình vẽ

-Các thành phần chính của Tkinter:

+Tk() – Tạo cửa sổ chính của ứng dụng

+Frame – Khung chứa các widget

+Label – Hiển thị văn bản hoặc hình ảnh

+Button – Tạo nút bấm tương tác

+Entry – Ô nhập văn bản

+Canvas – Vẽ hình ảnh hoặc đồ họa sử dụng cho giao diện bàn cờ của game Nối 4

+Grid, Pack, Place – Các phương pháp bố trí giao diện

## ***2.2.Thuật toán Minimax***

Thuật toán Minimax được sử dụng để mô phỏng các quyết định của AI trong game đối kháng hai người. Trong game “Nối 4”, AI sẽ sử dụng Minimax để tìm ra nước đi có lợi nhất nhằm giành chiến thắng hoặc ngăn chặn đối thủ

-Minimax hoạt động dựa trên nguyên lý rằng:

+Người chơi MAX cố gắng tối đa hóa lợi thế bằng cách chọn nước đi tốt nhất

+Người chơi MIN cố gắng tối thiểu hóa lợi thế của đối phương bằng cách chọn nước đi có điểm thấp nhất

-Cấu trúc thuật toán Minimax:

+Xây dựng cây trạng thái: Mỗi nút trong cây biểu diễn một trạng thái của bàn cờ sau khi một nước đi được thực hiện

+Đánh giá trạng thái: Dùng một hàm lượng giá để xác định điểm số cho từng trạng thái

-Duyệt cây trạng thái:

+Nếu là lượt của MAX, chọn nước đi có điểm cao nhất

+Nếu là lượt của MIN, chọn nước đi có điểm thấp nhất

-Quay lui (Backtracking): Trả về nước đi tốt nhất sau khi đã duyệt qua các trạng thái

-Hạn chế của Minimax: Nếu không có giới hạn về độ sâu tìm kiếm, Minimax có thể duyệt tất cả các trạng thái có thể xảy ra, dẫn đến tốc độ xử lý chậm. Khi số lượng trạng thái có thể xảy ra quá lớn, thuật toán sẽ tốn nhiều tài nguyên để tính toán. Để giải quyết vấn đề này, ta sử dụng cắt tỉa Alpha-Beta để giảm số lượng trạng thái cần duyệt

## **2.3.Cắt tỉa Alpha-Beta**

Cắt tỉa Alpha-Beta là một kỹ thuật tối ưu dành cho Minimax, giúp tăng tốc độ xử lý bằng cách loại bỏ những trạng thái không cần thiết

-Cơ chế hoạt động:

+Alpha (α): Điểm tốt nhất mà người chơi MAX có thể đảm bảo được

+Beta (β): Điểm tốt nhất mà người chơi MIN có thể đảm bảo được

+Nếu một trạng thái có giá trị kém hơn Beta, ta có thể bỏ qua các trạng thái con của nó (cắt tỉa)

+Nếu một trạng thái có giá trị lớn hơn Alpha, ta cũng có thể bỏ qua vì đối thủ sẽ không chọn nó

-Ứng dụng Minimax và Alpha-Beta trong game "Nối 4": AI sử dụng Minimax để phân tích từng nước đi khả thi. Cắt tỉa Alpha-Beta giúp giảm số trạng thái cần xét, tối ưu hiệu suất. Độ sâu tìm kiếm có thể được điều chỉnh để tăng giảm mức độ khó của AI. Kết hợp thêm heuristics (giá trị trọng số cho các vị trí quan trọng trên bàn cờ) giúp AI đưa ra quyết định tốt hơn

## **2.4.Lập trình hướng đối tượng (OOP) trong Python**

Lập trình hướng đối tượng (OOP) là phương pháp lập trình dựa trên đối tượng và lớp, giúp mã nguồn trở nên dễ quản lý và mở rộng hơn. Nhờ OOP, chương trình được phân chia thành nhiều module rõ ràng, dễ bảo trì và nâng cấp

-Các nguyên lý cơ bản:

+Đóng gói (Encapsulation): Giữ dữ liệu bên trong lớp, hạn chế truy cập trực tiếp

+Kế thừa (Inheritance): Cho phép lớp con kế thừa đặc điểm của lớp cha

+Đa hình (Polymorphism): Một phương thức có thể hoạt động theo nhiều cách khác nhau

+Trừu tượng hóa (Abstraction): Ẩn đi chi tiết không cần thiết, chỉ hiển thị những gì quan trọng

-Ứng dụng OOP trong game Nối 4:

+Lớp Board: Quản lý trạng thái bàn cờ

+Lớp Player: Đại diện cho người chơi

+Lớp AIPlayer: Xử lý AI bằng Minimax

+Lớp Game: Điều phối trò chơi

## **2.5.Design Pattern MVC cho game**

MVC (Model-View-Controller) là một Design Pattern phổ biến, giúp chia chương trình thành ba phần riêng biệt:

+Model (M) – Xử lý dữ liệu: Lưu trữ trạng thái game - bàn cờ, lượt chơi. Kiểm tra điều kiện thắng-thua

+View (V) – Hiển thị giao diện: Sử dụng Tkinter để hiển thị bàn cờ và các nút điều khiển. Cập nhật hiển thị sau mỗi lượt chơi

+Controller (C) – Điều khiển game: Nhận đầu vào từ người chơi. Gọi các phương thức từ Model để thay đổi trạng thái. Kết nối với View để hiển thị cập nhật

-Ứng dụng MVC trong game Nối 4:

+Model: Lớp Board chứa dữ liệu bàn cờ

+View: Lớp GameUI (sử dụng Tkinter)

+Controller: Xử lý lượt chơi, gọi A.I khi cần

-Ưu điểm của MVC: dễ bảo trì, tách biệt giữa giao diện và logic xử lý. Có thể thay đổi giao diện Tkinter – Pygame mà không ảnh hưởng đến logic

## **Chương 3.Thiết kế và xây dựng chương trình**

## **3.1.Kiến trúc chương trình**

- Giao diện chính (main window)

- Bàn cờ game: ma trận 2D

- Module kiểm tra thắng: kiểm tra hàng, cột, đường chéo

- Module AI: áp dụng thuật toán minimax

## **3.2.Các chức năng chính**

- reset\_board(): khởi tạo bàn chơi

- drop\_piece(row, col): đặt quân cờ vào vị trí

- check\_winner(): xác định người thắng

- ai\_move(): thực hiện nước đi của máy

## **3.3.Sơ đồ UML**

## **3.4.Thiết kế giao diện người dùng**

## **Chương 4.Thực nghiệm và kết luận**

## ***4.1.Kết quả thực nghiệm***

-Hình ảnh giao diện chạy game

A screenshot of a computer

AI-generated content may be incorrect.

## ***4.2.Đánh giá và cải tiến***

-Đánh giá chung

+Mức độ hoàn thiện: 90%

+Giao diện trực quan, chơi mượt mà

+Chế độ chơi với máy có thể lựa chọn độ khó (số lớp tìm kiếm)

-Hướng cải tiến: thêm âm thanh, lưu điểm cao, giao diện đẹp hơn bằng thư viện như Pygame hoặc Kivy

# **III.TÀI LIỆU THAM KHẢO**

1.Tkinter Documentation: https://docs.python.org/3/library/tk.html

2.Minimax Algorithm: https://en.wikipedia.org/wiki/Minimax

3.GitHub Repository: https://github.com/NakerTheFirst/Four-in-a-row