CS 4850 - Fall 2024

SP-14 GREEN - Novel Chess Game

Group Members:

Joshua Peeples, Matthew Corvacchioli, Allen Smith, Dylan Luong, Ashton Miller

Professor and Date:

Sharon Perry – December 2nd, 2024

Project Website:

https://sp-14-green.tiiny.site/

GitHub Repository:

https://github.com/S-14Chess-p/Senior-Project-ChessAI

Table of Contents

1. Introduction and Overview 3
2. Design Considerations 3
   2.1 Design Overview 3
   2.2. General Constraints 3
   2.3. Development Methods 3
3. Architectural Strategies 4
4. System Architecture 5
5. Detailed System Design 5
   5.1. Classification 5
   5.2. Definition 5
   5.3. Constraints 6
   5.4. Interface/Exports 6
6. External Interface Requirements 6
   6.1 User Interface Requirements 6
   6.2 Hardware Interface Requirements 6
7. Glossary 7
8. Bibliography 8

1. Introduction and Overview

The System Design-Documents (SDD) describes how all the different parts identified in the requirements document interact with each other at a high level. The SDD orchestrates our design goals clearly and will provide an overview of the system architecture. Lastly, the SDD describes the goals we had in mind when designing our product.

The SDD provides the development team guidance for how the system architecture and design should be set up. The intent is to provide the development team with documentation to refer to when they are stuck, as well as documentation to serve as a structural piece that helps with the whole team's understanding of the architecture of the project. These documents are meant to be at a high level and as such anyone other than those the SDD is meant for would have difficulty understanding them. As the development period of the project continues, the team will grow much more of a low-level understanding of the project's ins and outs based on the expansion of the high-level knowledge from this document.

2. Design Considerations

2.1 Design Overview

C# will be the primary language used to code the logic behind our project.

Html file will be used to set up the user interface for our project. This HTML file will contain the necessary CSS stylings for the front-end design that the project aims to meet spec-wise.

Windows will be the primary operating system used for this project.

The user will be utilizing our program to play and enjoy the game of chess either with a friend in the player vs player mode or against a trained AI that we have programmed to learn the game to a playable metric.

2.2. General Constraints

C# is server-side and as such is unable to run in a browser with an html file like a language like JavaScript would be able to, so we will use ASP.net framework that allows us to bypass this issue.

2.3. Development Methods

We will begin by developing the game of chess as everyone knows it in the player vs player form so that we know our program works. Once complete, we will alter it to use our own custom ruleset still focusing on the player vs player mode. Once we are happy with how that is functioning, we will begin to develop player vs AI and hopefully have a fully fleshed out product by the end and expand beyond that if we have the time to do so.

3. Architectural Strategies

Our goal is to use the C# language for the overall design of the game and the functionality of the games rules/AI as well as using HTML and CSS for the design of the frontend for our user interface. The IDE of choice will be Microsoft Visual Studio through our KSU accounts.

We currently do not have any plans to reuse any existing software components and plan to rely fully on our own developed software.

In the coming weeks, we plan to be in the development phase of the project to put the project into the implementation phase. Once implemented, numerous testing methods will be utilized to make sure that the program runs according to our design and how we see fit. During the testing phase we will take adequate measures to make sure the program is polished before the final submission.

The main paradigm that we aim to focus on is the Graphical User Interface that will be utilized for the front end of the project where we will have a home screen, which will have 3 screens for navigation, the first screen will be a home screen providing a generic overview of the project. The second screen will navigate the user to a page that has a list of extensive rules, the third page will facilitate different versions of the game.

The first version of the game will allow two players to play against each other online on separate devices.

The second version of the game is a local player vs player version of the game where two players can simultaneously play against one another.

The third version of the game is a player vs AI version of the game where the user can play against an AI that is engineered with different AI heuristics.

As mentioned before each of these modes will be readily accessible from the play screen that the user can easily navigate to from the Home screen.

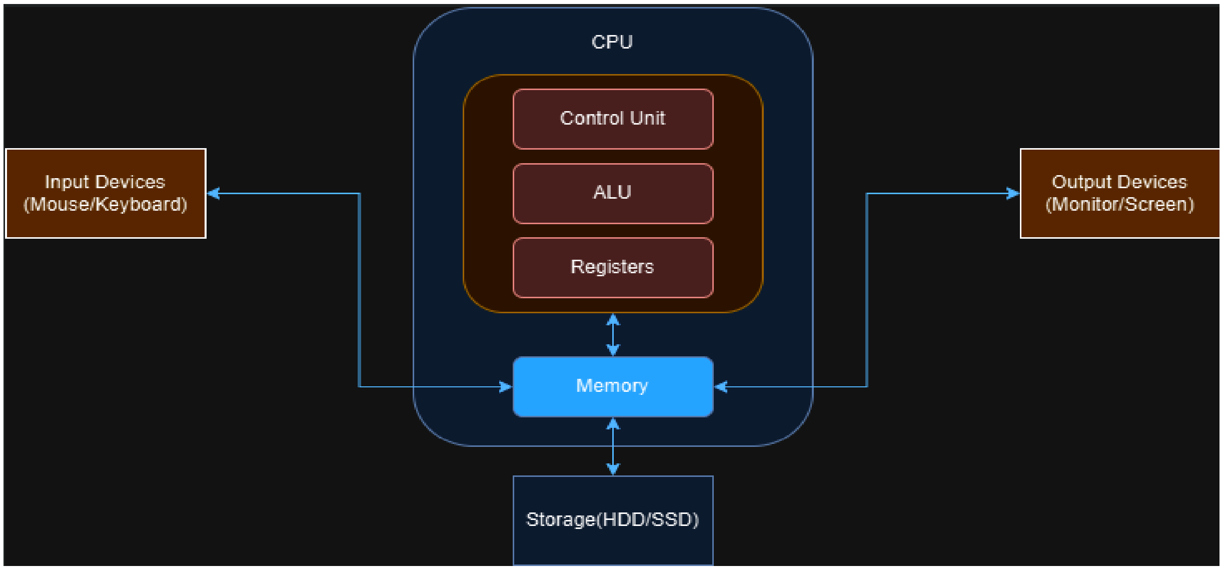
The programming language that we intend to use to develop the program is the C# programming language. Alongside C#, we intend to use the HTML markup language along with CSS styling to refine the design of the programs front end. The ASP.net framework will be used to allow these two file types to work with each other.

We will use the exception error handling method.

4. System Architecture

We hope for the system architecture to remain simplistic and will always be willing to expand it, if necessary, but as it stands right now C# will be the primary back-end language responsible for running the AI as well as the logic for detecting valid moves. That C# code will then need to run through a web Framework that will allow it to communicate with an html file, which will be responsible for the front-end UI as well as sending user input back to the C# program for detecting valid moves.

Below, a system architecture design diagram can be found.


CPU
Control Unit
ALU
Registers
Memory
Input Devices (Mouse/Keyboard)
Output Devices (Monitor/Screen)
Storage(HDD/SSD)

5. Detailed System Design

5.1. Classification

The Minimax algorithm will be used as the primary component behind the AI; it utilizes a heuristic to make an informed decision on the best move available. This algorithm allows the programmer to limit how far in advance The AI can look, lowering the capabilities of the AI so it does not make the perfect move every time.

5.2. Definition

Minimax algorithm - Common in 2 player turn based games, the minimax algorithm represents the game in a partial heuristic, which uses the current state of the board as the root. The algorithm begins by generating all the possible moves for a defined depth which limits how far ahead the AI can look. This depth value helps the performance of the AI and making the AI to not make the perfect choice every time. Once the heuristic is fully generated, the AI uses some logic to determine the best possible move with the generated move list.

5.3. Constraints

C# is server-side and therefore requires a framework to work with html. We will be using ASP.net to handle this.

To keep the AI from slowing the game down we will have to test different depths within the minimax algorithm to make an effective AI without hindering performance.

5.4. Interface/Exports

The set of services (resources, data, types, constants, subroutines, and exceptions) that are provided by this component. The precise definition or declaration of each such element should be present, along with comments or annotations describing the meanings of values, parameters, etc.

For each service element described, include (or provide a reference) in its discussion a description of its important software component attributes (Classification, Definition, Responsibilities, Constraints, Composition, Uses, Resources, Processing, and Interface).

6. External Interface Requirements

6.1 User Interface Requirements

Three pages labeled: Home, Rules, and Play. Home will give a general overview of the project. Rules will give a non-visual written version of the ruleset. When the Play button is selected, the user is presented with three options: Online Player vs Player, Local Player vs player, and Player vs AI. All three will present the user with a chess board to play with the option they select just changes how the back end will function. Player vs Player and Player vs AI will each have a UI element indicating to the user which mode has been selected, but there will be no additional UI elements other than that.

6.2 Hardware Interface Requirements

6.2.1 Display Interface

A screen or monitor displays the game board, pieces, and user interface elements. At least a basic LED display will suffice.

6.2.2 User Input Interface

A mouse and keyboard will be used to support the chess application on a computer. Input will be through a mouse for selecting and moving pieces on the board, and a keyboard for entering settings.

6.2.3 Battery

The hardware should have enough battery life to support the game.

6.2.4 Processing Hardware

Central Processing Unit (CPU), Graphics Processing Unit (GPU), and Memory are required to play the game.

7. Glossary

C#- An object-oriented programming language that is utilized primarily for backend development.

HTML - A markup language that is utilized for frontend development.

CSS - A styling language that is used to structure the developed frontend.

ASP.net framework - A popular web framework that allows developers to design and develop web applications and select services on the web.

IDE - Integrated Development Environment

Visual Studio - One of many industries standard IDEs utilized in several different kinds of projects.

Minimax algorithm - An algorithm that uses a heuristic to make a choice for a solution that best fits the given scenario that it is presented.

Heuristic - A form of Artificial Intelligence that combines a weighted tree and search algorithm to determine the best possible move.

8. Bibliography

GeeksforGeeks. (2019, November 14). *Introduction to ASP.NET*. https://www.geeksforgeeks.org/introduction-to-asp-net/

GeeksforGeeks. (2022, June 13). *Minimax Algorithm in Game Theory* | *Set 1 (Introduction)*. https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/