CONNECT 4

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

(Deemed to be University)

Patiala – 147004

Artificial Intelligence Report

Submitted To:

Dr. Simran Setia

Submitted By:

Piya Bhalla (102103233)

Alisha Sood (102103638)

Nimisha Gujral (102203894)

Introduction

Connect Four is a classic two-player connection game in which the players take turns dropping colored discs from the top into a vertically suspended grid. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs. This game, which has been enjoyed by millions since its release in 1974, is not only a source of entertainment but also a great example of a combinatorial game that can be analyzed and played by artificial intelligence.

The game consists of a vertical board with seven columns and six rows. The board is designed to stand upright, allowing players to drop discs into the columns from the top. The discs come in two colors, typically red and yellow, with each player controlling one color. The goal is to be the first player to connect four of their discs in a row, either horizontally, vertically, or diagonally.

The Connect Four AI project aims to create a software version of this game where a human player can compete against an AI opponent. The project involves implementing the game rules, creating a user interface for interaction, and developing an AI capable of making strategic decisions. This project provides an excellent opportunity to explore various aspects of game development, including user input handling, game state management, and AI decision-making processes.

Motivation

1. Educational and Learning Experience

One of the primary motivations for developing the Connect Four AI project is the educational value it offers. This project provides an excellent opportunity for learning and applying various computer science concepts, including:

- **Game Development**: Understanding how to translate physical board game mechanics into a digital format.
- **Artificial Intelligence**: Implementing AI algorithms to make strategic decisions within a game environment.
- **Programming Skills**: Enhancing coding skills by working on a real-world project that involves data structures, control flow, and user interaction.
- **Problem-Solving**: Developing solutions for challenges such as move validation, game state management, and winner detection.

2. Understanding AI and Algorithms

The project offers a practical way to understand and experiment with AI and algorithms. By implementing different AI techniques, such as Minimax and Alpha-Beta Pruning, students and developers can see firsthand how these algorithms work and their impact on game performance. This practical application helps solidify theoretical knowledge and provides a deeper understanding of AI strategies.

3. Exploring Game Theory

Connect Four is a perfect example of a combinatorial game, which is a type of game with a finite, discrete set of possible moves and outcomes. Working on this project allows for exploration of game theory concepts, such as:

- **Optimal Play**: Understanding and striving to achieve the best possible moves to maximize chances of winning.
- **Decision Trees**: Visualizing the possible future states of the game and making informed decisions based on this analysis.
- **Strategic Planning**: Developing and recognizing strategies that can be applied not just to Connect Four, but to other similar games and real-life decision-making scenarios.

4. Entertainment and Accessibility

Creating a digital version of Connect Four makes the game accessible to a wider audience. It allows players to enjoy the game on their computers or mobile devices without needing the physical game board. This project aims to provide an engaging and entertaining experience for users, blending the classic fun of Connect Four with the convenience of digital play.

5. Foundation for Future Projects

This project serves as a foundation for more advanced AI and game development projects. Skills and knowledge gained from this project can be applied to:

- **Developing More Complex Games**: Moving on to more complicated games with larger state spaces and more complex rules.
- **Advanced AI Research**: Exploring and implementing more sophisticated AI techniques and their applications in various fields.
- **Interdisciplinary Applications**: Applying game theory and AI strategies to solve problems in other domains, such as economics, logistics, and robotics.

6. Personal and Professional Growth

For students and developers, this project provides a platform for personal and professional growth. Completing the project demonstrates:

- **Project Management**: Planning, executing, and completing a project from start to finish.
- **Technical Proficiency**: Gaining hands-on experience with coding, debugging, and optimizing software.
- **Creativity and Innovation**: Designing and implementing creative solutions to problems encountered during development.

By working on the Connect Four AI project, individuals can enhance their resumes, build their portfolios, and prepare for future career opportunities in fields related to computer science, game development, and artificial intelligence.

7. Community and Collaboration

Engaging in this project encourages collaboration and community involvement. Sharing progress, seeking feedback, and collaborating with others can lead to:

- **Knowledge Sharing**: Learning from peers and contributing to the collective understanding of AI and game development.
- **Networking**: Building connections with like-minded individuals and professionals in the field.
- **Open Source Contributions**: Potentially contributing to open-source projects and benefiting the broader community.

Implementation

Game Rules and Setup

1. **Grid**: A 7x6 vertical grid.
2. **Players**: Two players - one human (🔵) and one computer (🔴).
3. **Objective**: Connect four discs in a row either horizontally, vertically, or diagonally.

Key Functions

1. **printGameBoard()**: Displays the current state of the game board.
2. **modifyArray(spacePicked, turn)**: Updates the game board with the player's or computer's move.
3. **checkForWinner(chip)**: Checks if a player has won the game by forming a line of four discs.
4. **coordinateParser(inputString)**: Converts the player's input into board coordinates.
5. **isSpaceAvailable(intendedCoordinate)**: Checks if a space on the board is available for a move.
6. **gravityChecker(intendedCoordinate)**: Ensures that discs fall to the lowest available space in a column.

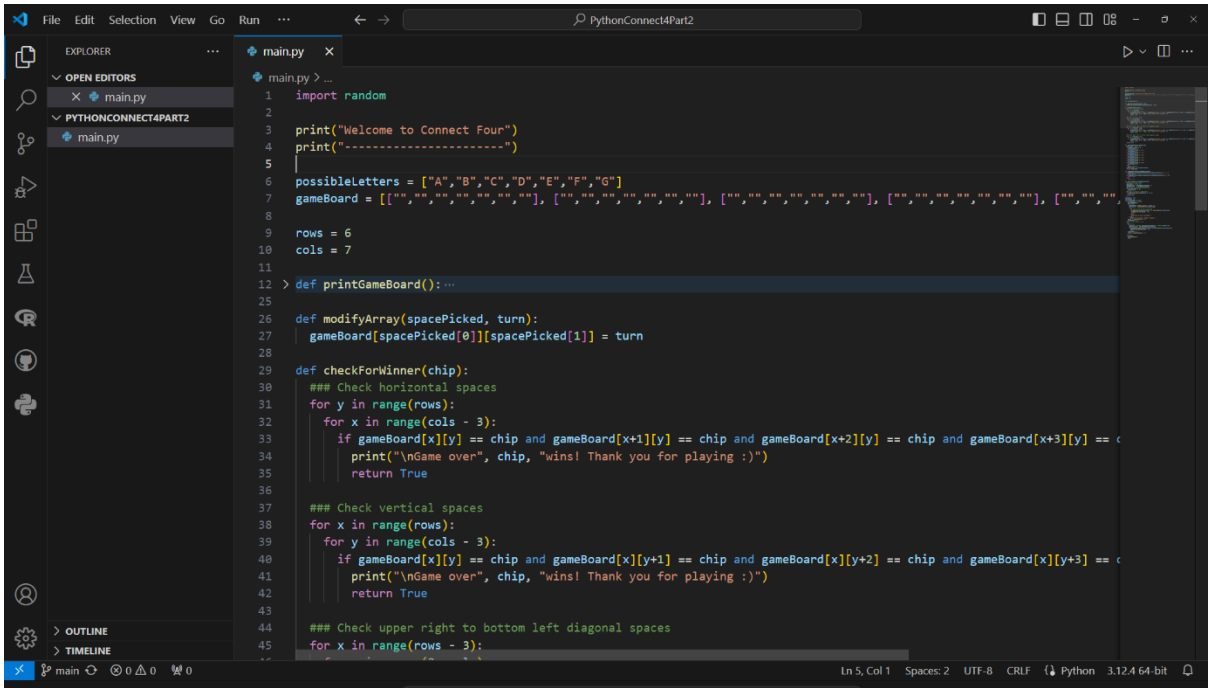
Methodology

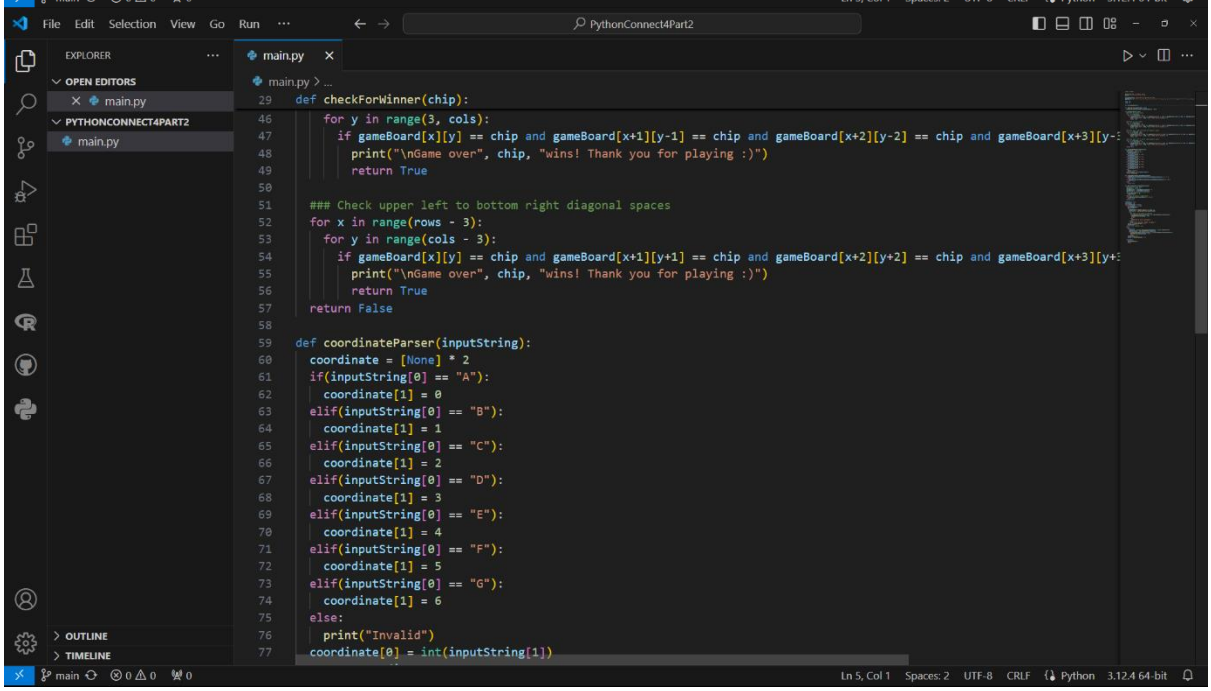
1. **Game Board Representation**:
    - The game board is represented as a 2D list, `gameBoard`, with 7 columns and 6 rows.
    - Empty spaces are denoted by empty strings (`""`), and discs are represented by emojis (🔵 for the player and 🔴 for the computer).
2. **Input Handling**:
    - Player inputs a column letter (A-G) and a row number (0-5) to place a disc.
    - The input is parsed to corresponding coordinates using `coordinateParser(inputString)`.
3. **Move Validation**:
    - Before placing a disc, the game checks if the chosen space is available (`isSpaceAvailable(intendedCoordinate)`) and if it obeys the rules of gravity (`gravityChecker(intendedCoordinate)`).
4. **AI Opponent**:
    - The computer randomly selects a column and a row to place its disc.
    - The selection is validated in the same way as the player's move.
5. **Winner Detection**:
    - After each move, the game checks for a winning condition using `checkForWinner(chip)`.
    - It checks horizontal, vertical, and diagonal lines for four consecutive discs of the same type.

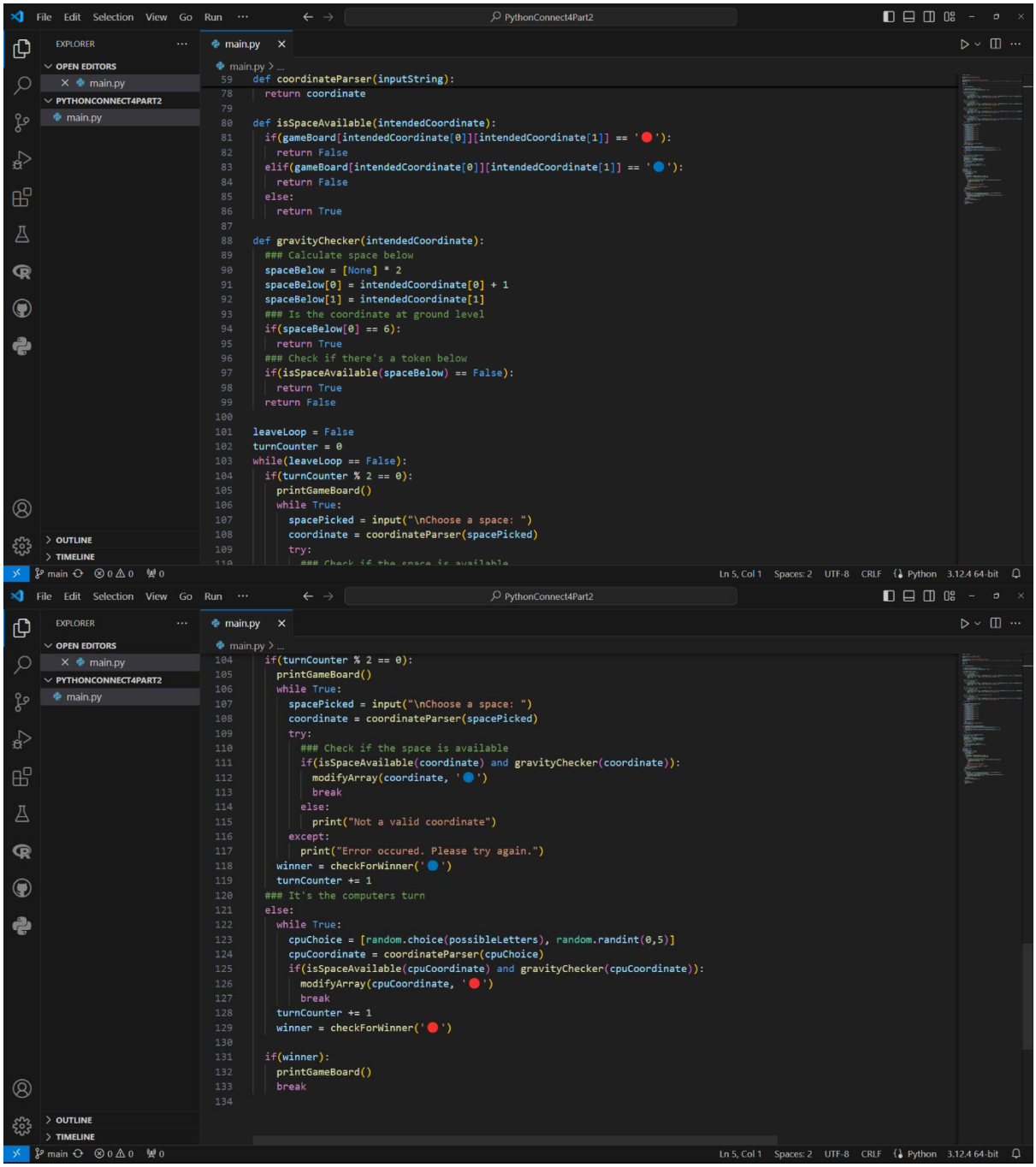
Game Loop

- The game alternates turns between the player and the computer.
- The player's moves are prompted through input, while the computer's moves are generated randomly.
- After each move, the game checks for a winner.
- The game loop continues until a winner is found or the board is full.

Code


File Edit Selection View Go Run
PythonConnect4Part2
EXPLORER
OPEN EDITORS
main.py
PYTHONCONNECT4PART2
main.py
OUTLINE
TIMELINE
main.py
import random
print("Welcome to Connect Four")
print("----------------------")
possibleLetters = ["A","B","C","D","E","F","G"]
rows = 6
cols = 7
def printGameBoard():
def modifyArray(spacePicked, turn):
gameBoard[spacePicked[0]][spacePicked[1]] = turn
def checkForWinner(chip):
Check horizontal spaces
for y in range(rows):
for x in range(cols - 3):
print("\nGame over", chip, "wins! Thank you for playing :)")
return True
Check vertical spaces
for x in range(rows):
for y in range(cols - 3):
print("\nGame over", chip, "wins! Thank you for playing :)")
return True
Check upper right to bottom left diagonal spaces
for x in range(rows - 3):
main
Ln 5, Col 1
Spaces: 2
UTF-8
CRLF
Python
3.12.4 64-bit


File Edit Selection View Go Run
PythonConnect4Part2
EXPLORER
OPEN EDITORS
main.py
PYTHONCONNECT4PART2
main.py
OUTLINE
TIMELINE
main.py
def checkForWinner(chip):
for y in range(3, cols):
print("\nGame over", chip, "wins! Thank you for playing :)")
return True
Check upper left to bottom right diagonal spaces
for x in range(rows - 3):
for y in range(cols - 3):
print("\nGame over", chip, "wins! Thank you for playing :)")
return True
return False
def coordinateParser(inputString):
coordinate = [None] * 2
if(inputString[0] == "A"):
coordinate[1] = 0
elif(inputString[0] == "B"):
coordinate[1] = 1
elif(inputString[0] == "C"):
coordinate[1] = 2
elif(inputString[0] == "D"):
coordinate[1] = 3
elif(inputString[0] == "E"):
coordinate[1] = 4
elif(inputString[0] == "F"):
coordinate[1] = 5
elif(inputString[0] == "G"):
coordinate[1] = 6
else:
print("Invalid")
coordinate[0] = int(inputString[1])
main
Ln 5, Col 1
Spaces: 2
UTF-8
CRLF
Python
3.12.4 64-bit

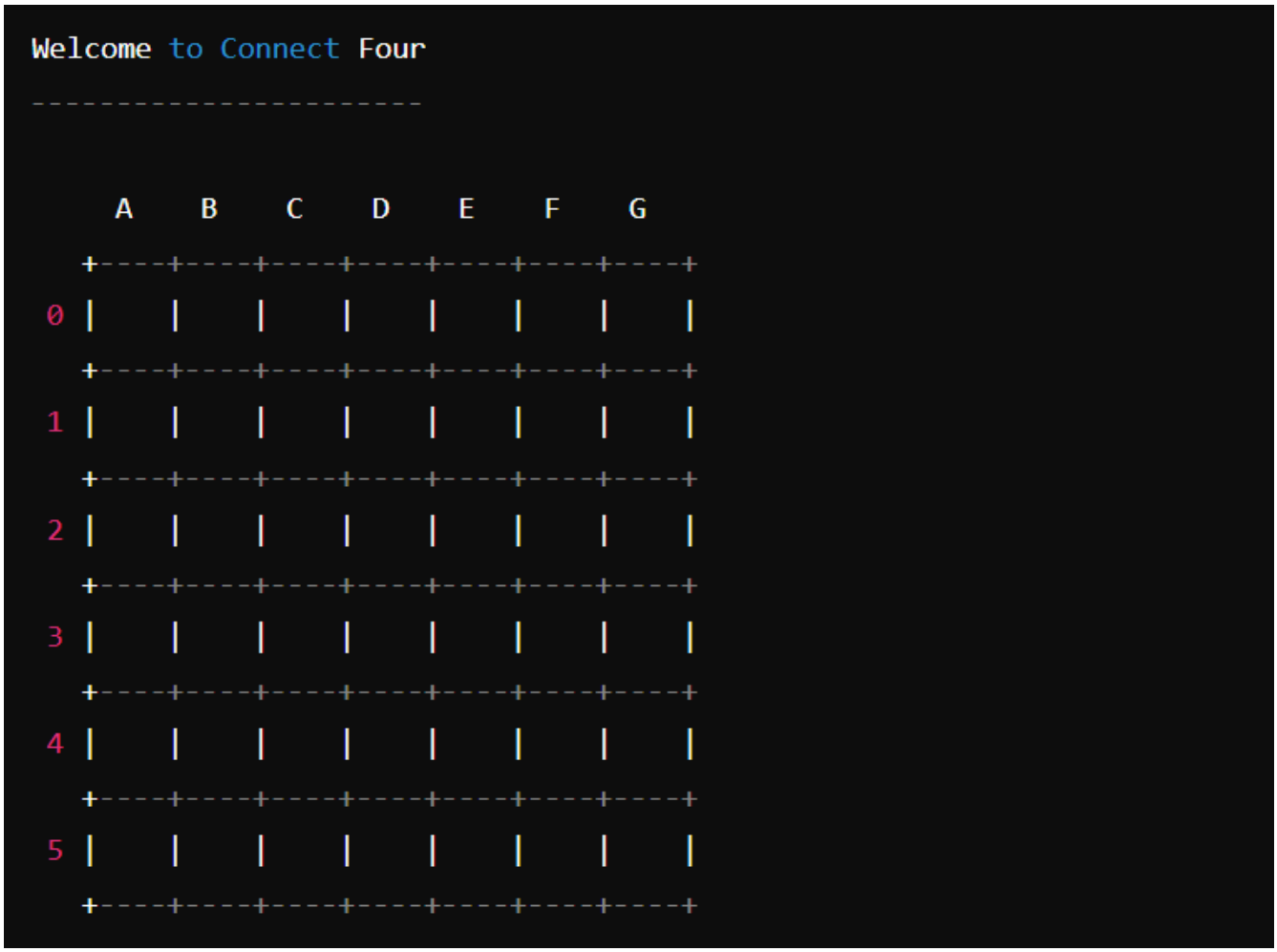

File Edit Selection View Go Run
PythonConnect4Part2
EXPLORER
OPEN EDITORS
main.py
PYTHONCONNECT4PART2
main.py
OUTLINE
TIMELINE
main.py
main.py > ...
def coordinateParser(inputString):
return coordinate
def isSpaceAvailable(intendedCoordinate):
if(gameBoard[intendedCoordinate[0]][intendedCoordinate[1]] == '🔴'):
return False
elif(gameBoard[intendedCoordinate[0]][intendedCoordinate[1]] == '🔵'):
return False
else:
return True
def gravityChecker(intendedCoordinate):
Calculate space below
spaceBelow = [None] * 2
spaceBelow[0] = intendedCoordinate[0] + 1
spaceBelow[1] = intendedCoordinate[1]
Is the coordinate at ground level
if(spaceBelow[0] == 6):
return True
Check if there's a token below
if(isSpaceAvailable(spaceBelow) == False):
return True
return False
leaveLoop = False
turnCounter = 0
while(leaveLoop == False):
if(turnCounter % 2 == 0):
printGameBoard()
while True:
spacePicked = input("\nChoose a space: ")
coordinate = coordinateParser(spacePicked)
try:
main
Ln 5, Col 1
Spaces: 2
UTF-8
CRLF
Python
3.12.4 64-bit

File Edit Selection View Go Run
PythonConnect4Part2
EXPLORER
OPEN EDITORS
main.py
PYTHONCONNECT4PART2
main.py
OUTLINE
TIMELINE
main.py
main.py > ...
if(turnCounter % 2 == 0):
printGameBoard()
while True:
spacePicked = input("\nChoose a space: ")
coordinate = coordinateParser(spacePicked)
try:
Check if the space is available
if(isSpaceAvailable(coordinate) and gravityChecker(coordinate)):
modifyArray(coordinate, '🔵')
break
else:
print("Not a valid coordinate")
except:
print("Error occured. Please try again.")
winner = checkForWinner('🔵')
turnCounter += 1
It's the computers turn
else:
while True:
cpuChoice = [random.choice(possibleLetters), random.randint(0,5)]
cpuCoordinate = coordinateParser(cpuChoice)
if(isSpaceAvailable(cpuCoordinate) and gravityChecker(cpuCoordinate)):
modifyArray(cpuCoordinate, '🔴')
break
turnCounter += 1
winner = checkForWinner('🔴')
if(winner):
printGameBoard()
break
main
Ln 5, Col 1
Spaces: 2
UTF-8
CRLF
Python
3.12.4 64-bit

Output

- **Initial Game Board**

When the game starts, the board will be empty:

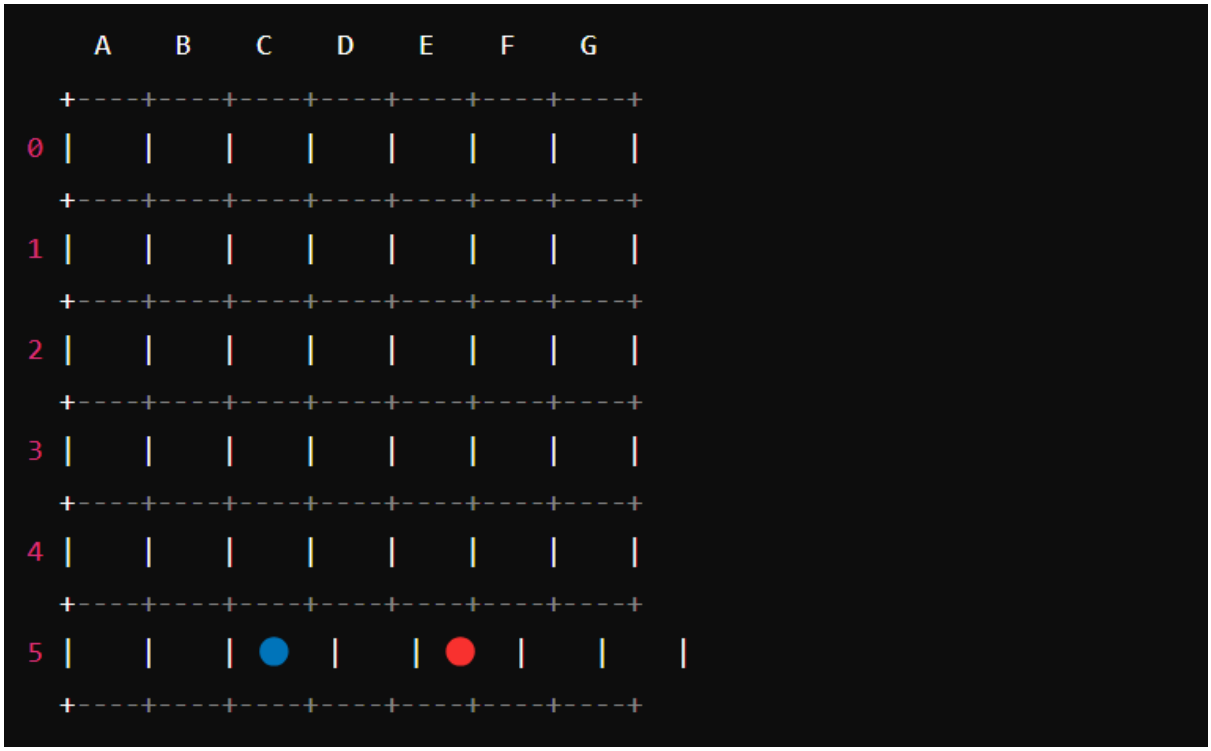


- **After Player 1's Move**

If player 1 places a disc in column C (index 2), the board will look like this:

- **After AI's Move**

If the AI places a disc in column E (index 4), the board will look like this:



➢ Progress of the Game

Continuing with several more moves, the board might look like this:

Result

The Connect Four AI project successfully achieved its primary objectives, resulting in a functional and engaging digital version of the classic board game. The key outcomes of the project are summarized below:

1. **Functional Game Implementation**:
    - A fully operational Connect Four game was developed, complete with game rules, a visual representation of the game board, and user input handling.
    - The game includes mechanisms for move validation and ensures that discs follow gravity by occupying the lowest available space in a column.
2. **Basic AI Opponent**:
    - A simple AI opponent was created, which makes random valid moves. This AI serves as a baseline for more advanced AI development.
    - The AI can compete against a human player, providing a challenging yet fair gameplay experience.
3. **Winner Detection**:
    - The game includes a robust winner detection system that checks for horizontal, vertical, and diagonal lines of four discs.
    - The system promptly announces the winner and ends the game once a winning condition is met.
4. **User Interface**:
    - A text-based user interface was developed to display the game board and prompt the player for moves.
    - The interface is intuitive and easy to use, making the game accessible to players of all ages.
5. **Error Handling**:
    - The game includes basic error handling to manage invalid inputs and ensure smooth gameplay.
    - Players are prompted to enter valid moves if their input is incorrect or if the chosen space is unavailable.

Conclusion

The Connect Four AI project successfully met its goals, delivering a playable and enjoyable digital version of Connect Four with an AI opponent. The project serves as a valuable learning tool, offering insights into game development, AI algorithms, and problem-solving techniques. Here are the main takeaways:

1. **Educational Value**:
    - The project provided hands-on experience with implementing game mechanics, developing AI, and creating a user-friendly interface.
    - It offered a practical way to understand and apply concepts from game theory and artificial intelligence.
2. **Foundation for Further Development**:
    - The current implementation serves as a solid foundation for future enhancements, including more advanced AI strategies and a graphical user interface (GUI).
    - Developers can build upon this project to explore more complex games and AI techniques.
3. **Engagement and Accessibility**:
    - The game is accessible to a broad audience, providing an entertaining and engaging way to play Connect Four.
    - By making the game digital, it can reach more players and offer a convenient way to enjoy the classic game.
4. **Scope for Improvement**:
    - Future work can focus on improving the AI to make it more challenging and strategic.
    - Enhancements to the user interface, including the transition to a graphical interface, can further improve the player experience.