



THE UNIVERSITY
of ADELAIDE

ARTIFICIAL INTELLIGENCE VS HUMAN : WHO WINS THE BOARD GAME?

PRESENTATION BY:

1. Vigneshraj Perumal Raja (a1787474)
2. Zichen Hao (a1751285)
3. Zhuoran Yang (a1779565)

UNDER THE GUIDANCE OF Peng Shi (Supervisor), Yang Fei
(Technical Advisor), Xin Yuan (Technical Advisor)

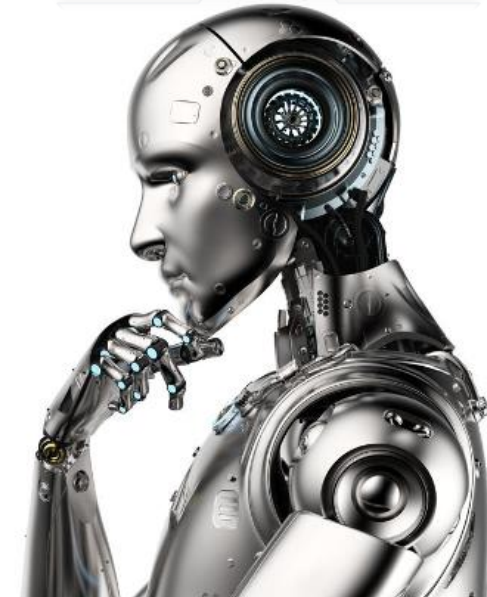
adelaide.edu.au

CONTENTS

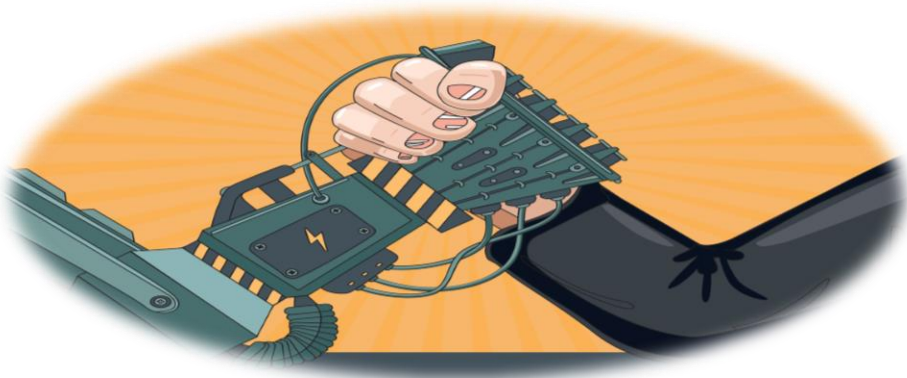
- ❖ Introduction
- ❖ Project Methodology
- ❖ Experimentation Methodology
- ❖ Results and Evaluation
- ❖ Conclusion
- ❖ References

Background

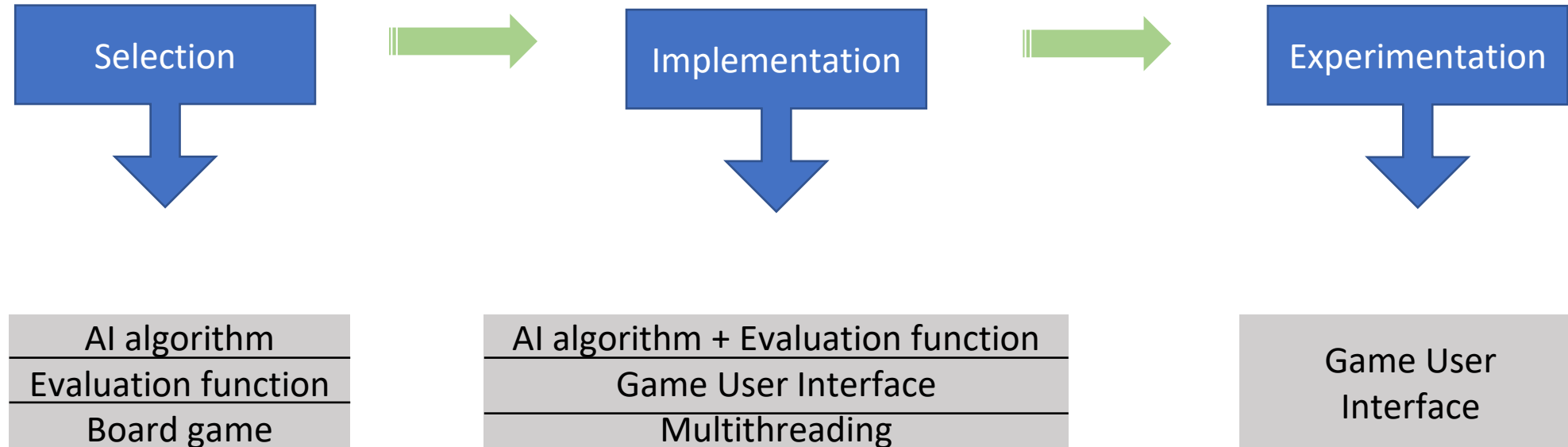
- ❖ **Artificial Intelligence** (AI) simulate and expand human's intelligence
- ❖ AI agent will be able to make self-governing decisions and produce AI functions independently
- ❖ Example: Robots, License Plate Recognition, etc
- ❖ Board game is a test-bed for AI algorithms
- ❖ Five in a row: put five pieces in a line



Aim & Objectives



- ❖ WHO WINS?, In the Board game battle
- ❖ To build a strong algorithm which helps AI but teaches human

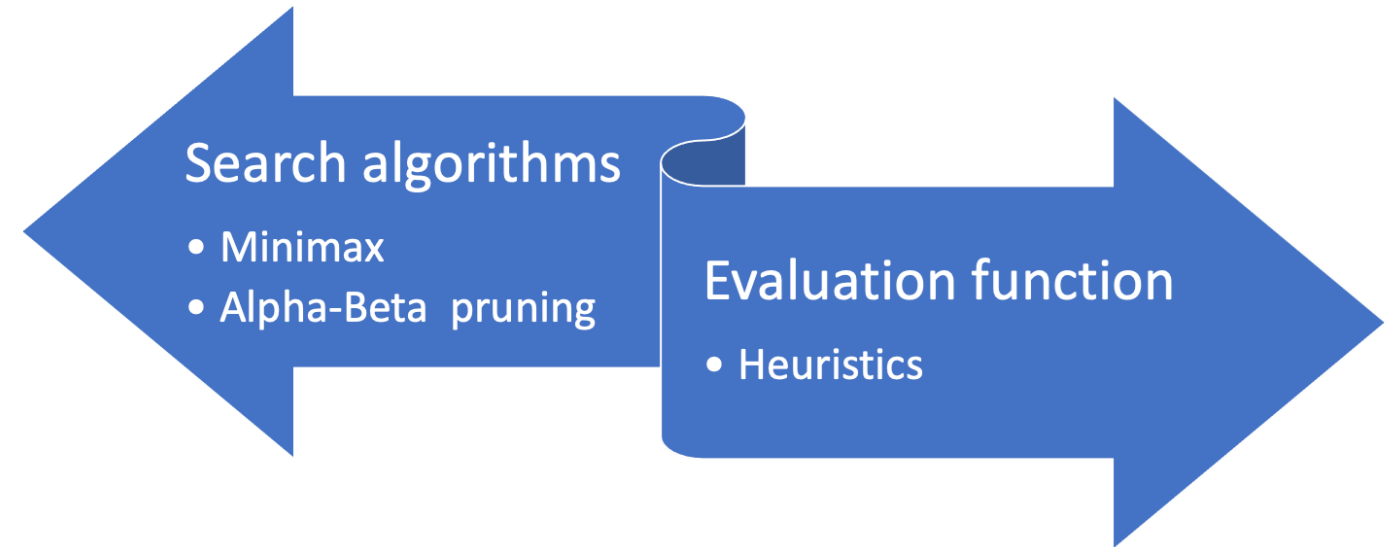


Technical Gap

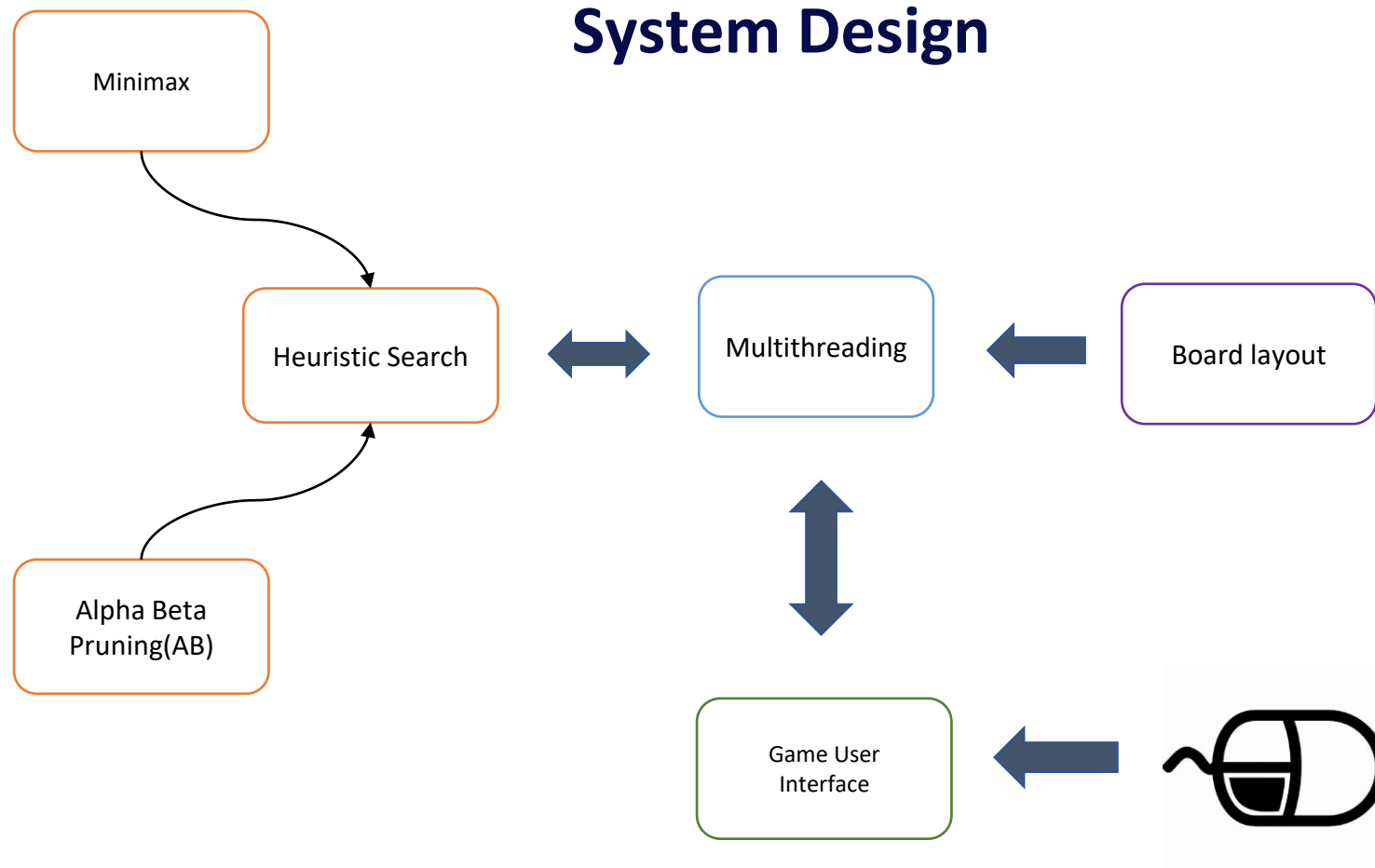


- ❖ Other gaming source wins human, does not help
- ❖ This project helps human by letting them know their wrong turn/decision
- ❖ Aims in developing a pro-player

Approach



System Design

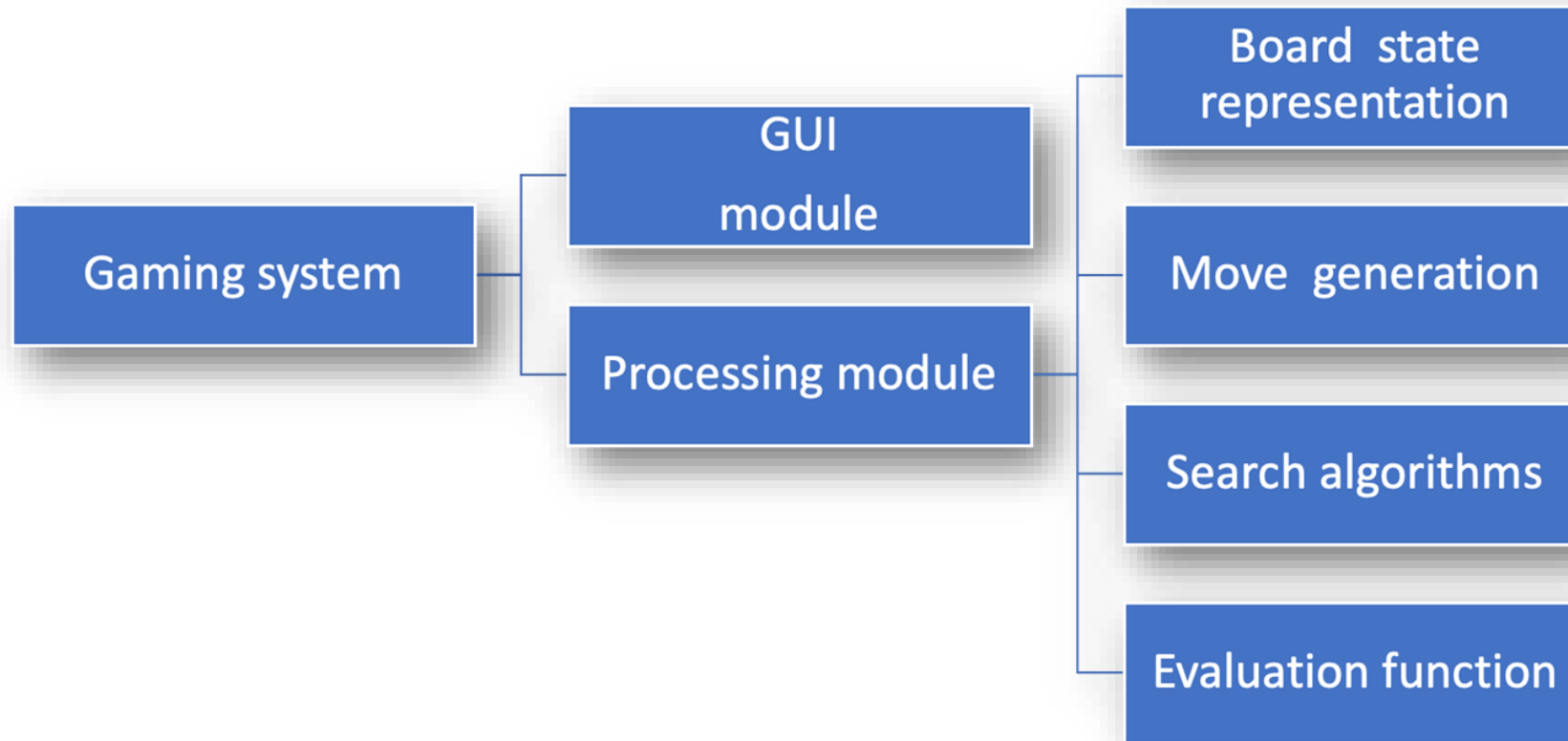


Selection

Implementation

Experimentation

Modules

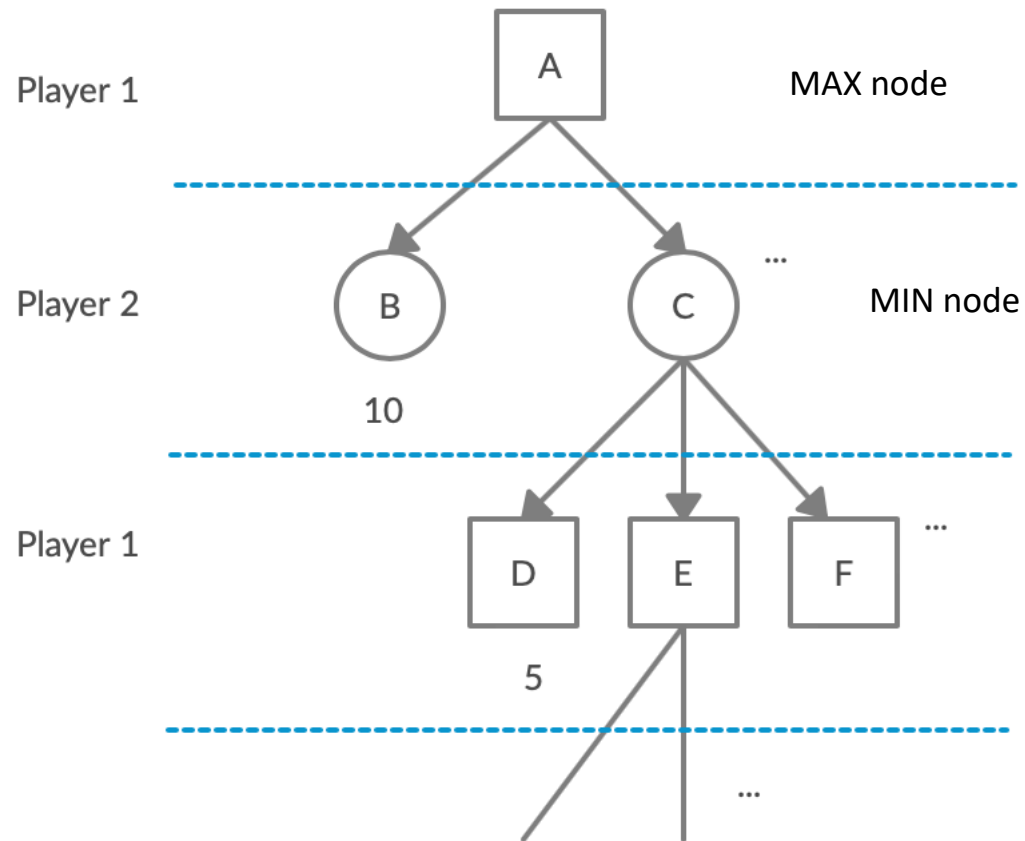


Selection

Implementation

Experimentation

Minimax Algorithm



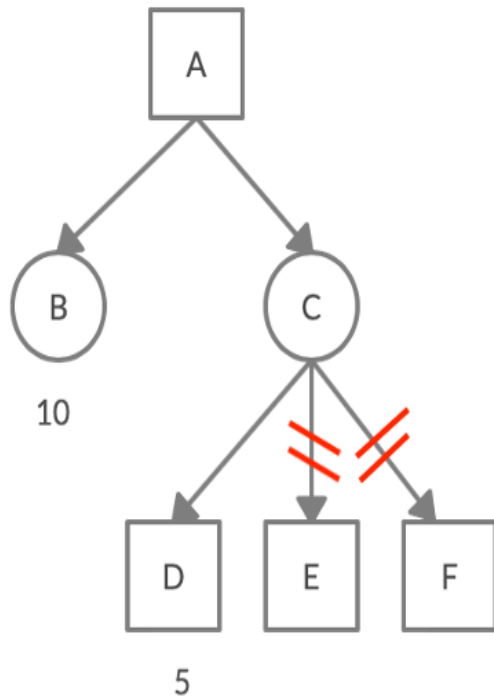
- ❖ Player 1 – AI – **Maximizes** its score
- ❖ Player 2 – Human player – **Minimises** AI's score
- ❖ Drawbacks: contain all possible moves including **unnecessary** branches.

Selection

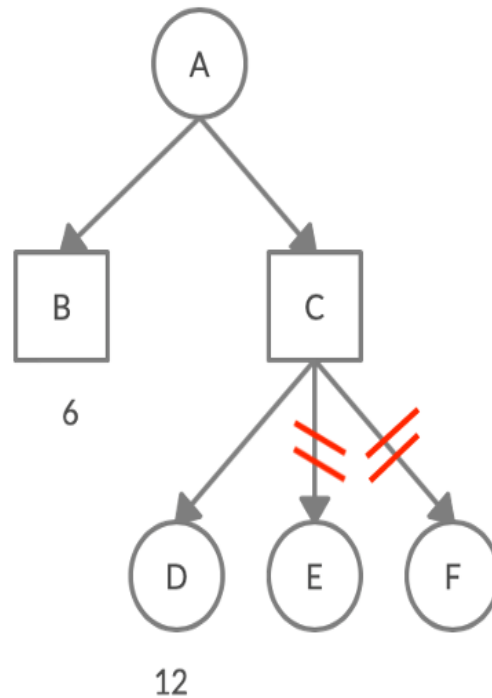
Implementation

Experimentation

Alpha Beta Pruning



(a) Alpha pruning



(b) Beta pruning

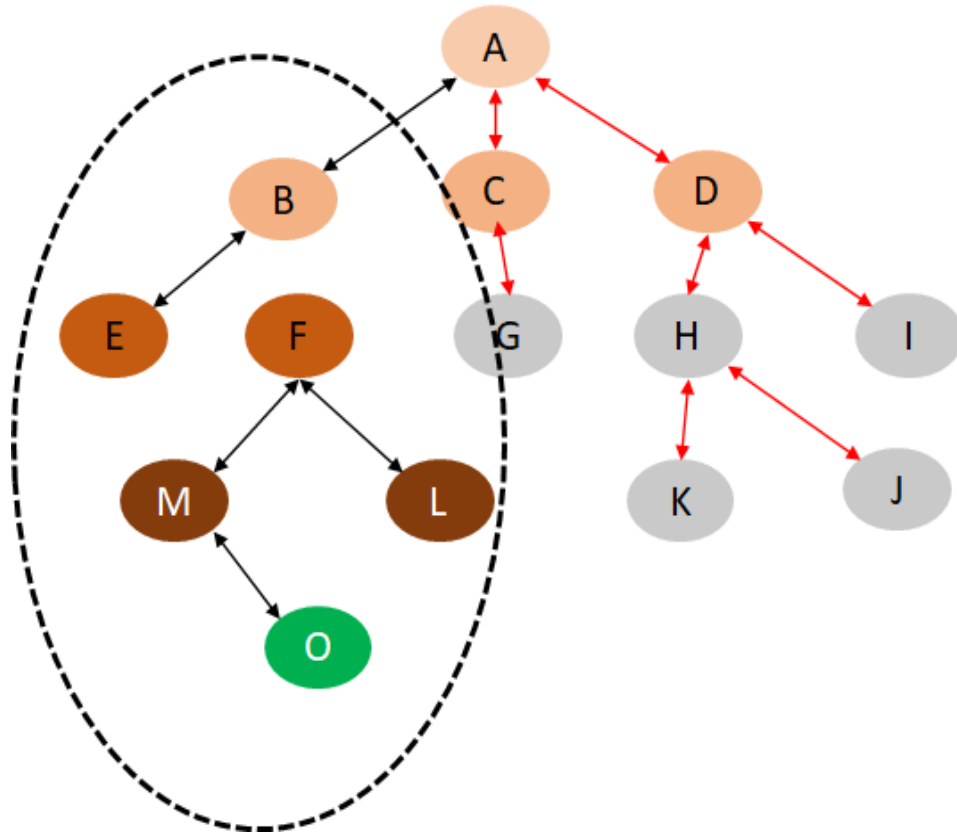
- ❖ Alpha pruning
- ❖ Beta pruning
- ❖ Drawback: **the order of nodes** greatly determines the efficiency.

Selection

Implementation

Experimentation

Evaluation Function



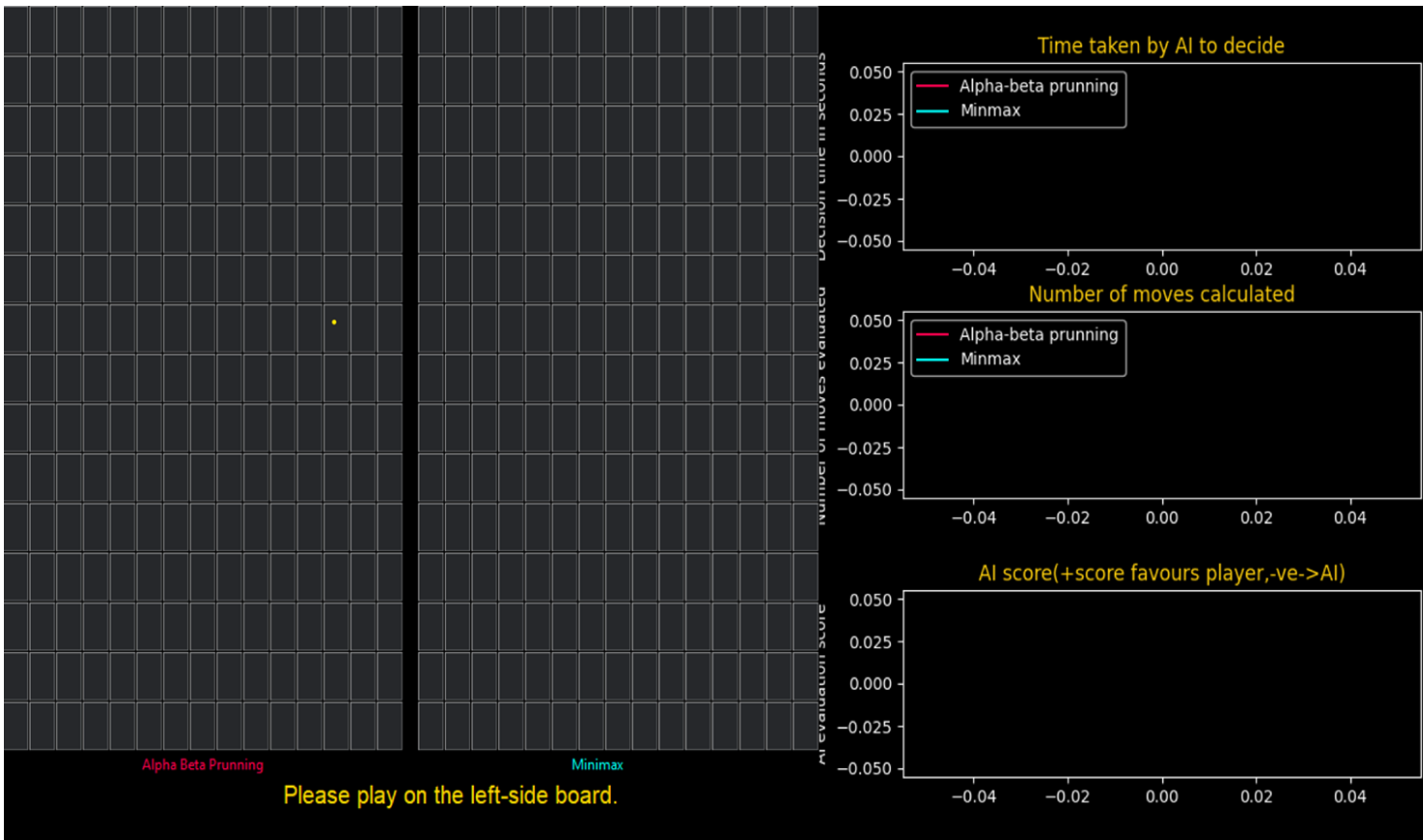
- ❖ Heuristic Search Function
- ❖ **Cut down** unnecessary nodes
- ❖ Reduces Evaluation time
- ❖ **Node Scorer**

Selection

Implementation

Experimentation

Graphical User Interface



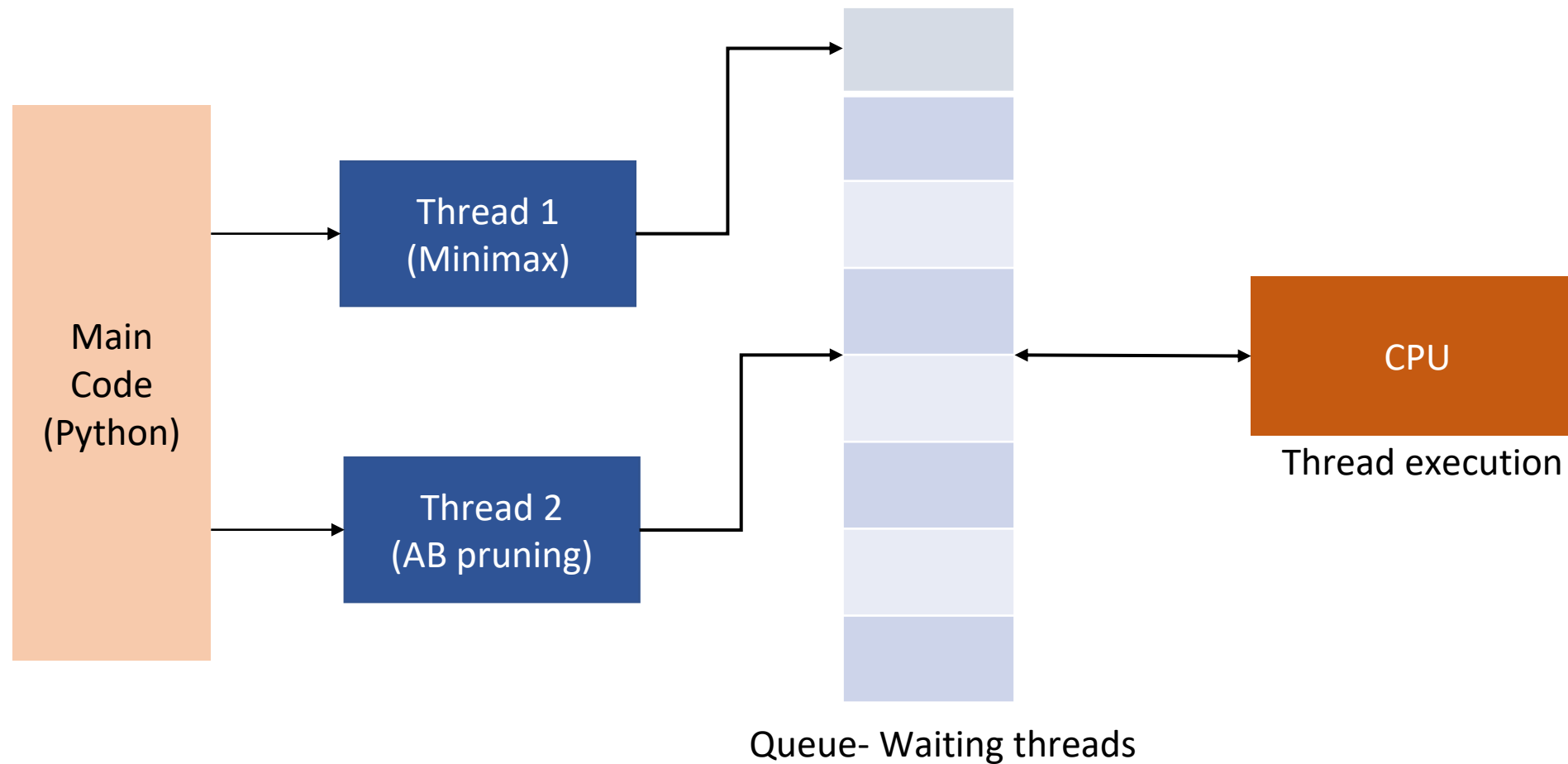
- ❖ A computer user interface display by graphic mode.
- ❖ **Man-machine** interaction
- ❖ Operability
- ❖ Practicability
- ❖ Technology

Selection

Implementation

Experimentation

Multithreading



Selection

Implementation

Experimentation

How we experimented?



Use one algorithm



Set the time and record



Ask different people to play



Place on different position



Calculate the time different algorithm used



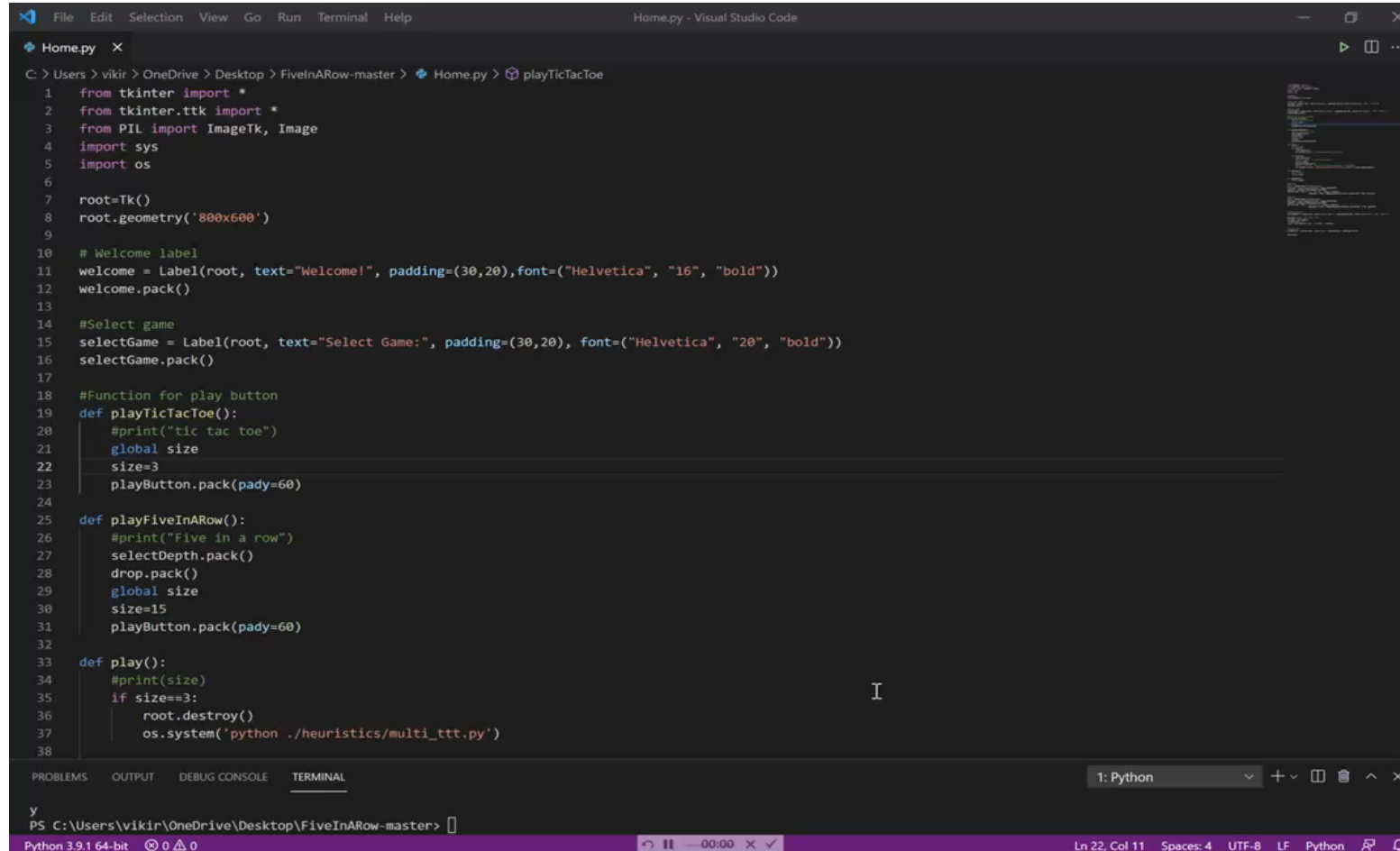
Compared the time

Selection

Implementation

Experimentation

Result



```
File Edit Selection View Go Run Terminal Help
Home.py - Visual Studio Code

Home.py X
C:\Users\vikir> OneDrive\ Desktop\ FiveInARow-master > Home.py > playTicTacToe

1 from tkinter import *
2 from tkinter.ttk import *
3 from PIL import ImageTk, Image
4 import sys
5 import os
6
7 root=Tk()
8 root.geometry('800x600')
9
10 # Welcome label
11 welcome = Label(root, text="Welcome!", padding=(30,20),font=("Helvetica", "16", "bold"))
12 welcome.pack()
13
14 #Select game
15 selectGame = Label(root, text="Select Game:", padding=(30,20), font=("Helvetica", "20", "bold"))
16 selectGame.pack()
17
18 #Function for play button
19 def playTicTacToe():
20     #print("tic tac toe")
21     global size
22     size=3
23     playButton.pack(pady=60)
24
25 def playFiveInARow():
26     #print("Five in a row")
27     selectDepth.pack()
28     drop.pack()
29     global size
30     size=15
31     playButton.pack(pady=60)
32
33 def play():
34     #print(size)
35     if size==3:
36         root.destroy()
37         os.system('python ./heuristics/multi_ttt.py')
38
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Python
Python 3.9.1 64-bit
```

Selection

Implementation

Experimentation

Evaluation

Minimax:

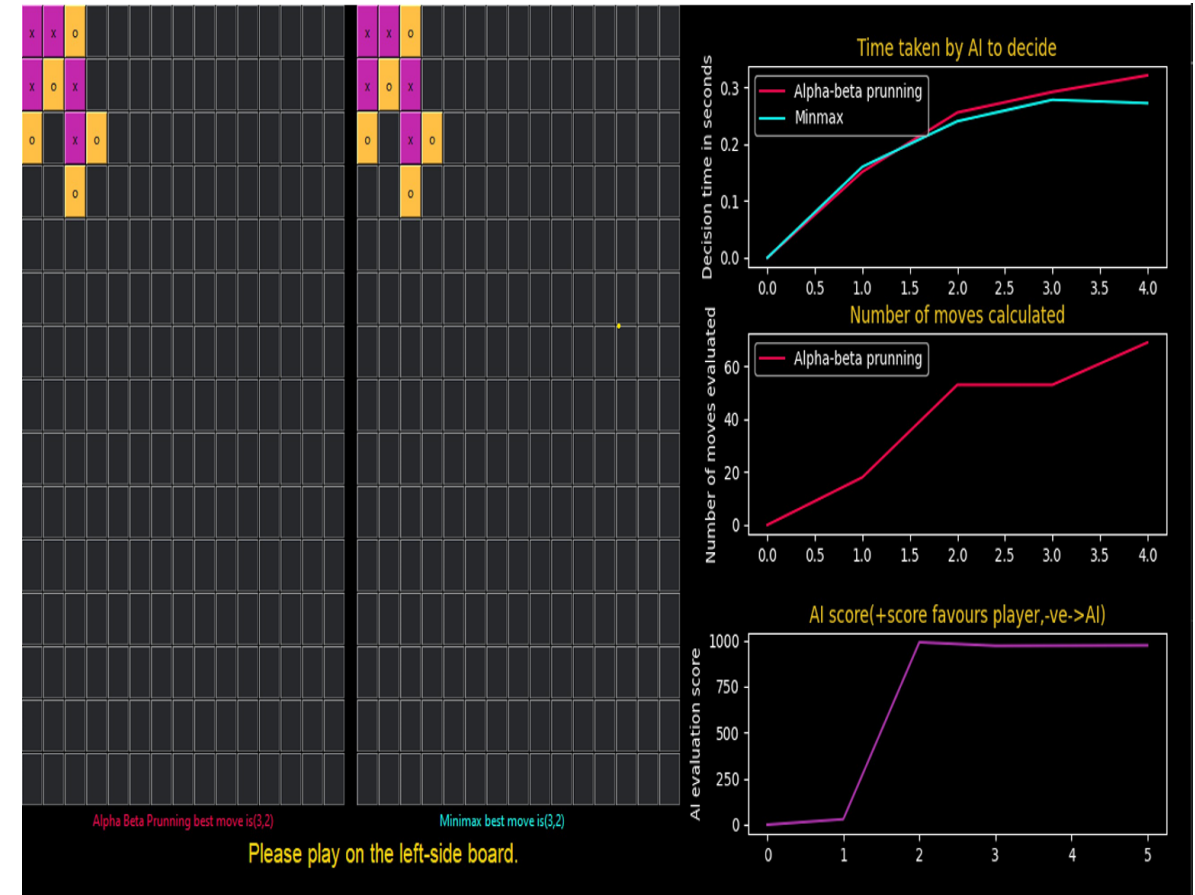
$$\text{Average time taken per move} = \frac{0.2 + 0.25 + 0.27 + 0.30 + 0.7}{5} \\ = 0.344 \text{ seconds}$$

$$\text{Average no.of moves per best move} = \frac{25+50+50+75+150}{5} \\ = 70 \text{ moves}$$

AB Pruning:

$$\text{Average time taken per move} = \frac{0.15 + 0.25 + 0.3 + 0.35 + 0.7}{4} \\ = 0.4375 \text{ seconds}$$

$$\text{Average number of moves per best move} = \frac{20+50+50+70}{4} \\ = 47.5 \text{ moves}$$



Selection

Implementation

Experimentation

Conclusion

With the help of Evaluation function, Search algorithms are able to win the Humans in the Five-in-a-row board game.



Selection

Implementation

Experimentation

References

- [1] Draskovic, D., M. Cvetanovic, and B. Nikolic, *SAIL—Software system for learning AI algorithms*. Computer applications in engineering education, 2018. **26**(5): p. 1195-1216.
- [2] Tzung-Pei, H., H. Ke-Yuan, and L. Wen-Yang, *A genetic minimax game-playing strategy*. 1998, IEEE. p. 690-694.
- [3] Venkateswara Reddy, L., et al., *Design and development of artificial intelligence (AI) based board game (Gobang) using android*. Materials today : proceedings, 2021.
- [4] Halim, Z., *Evolutionary Search in the Space of Rules for Creation of New Two-Player Board Games*. 2014.
- [5] Edelkamp, S. and S. Schroedl, *Heuristic Search : Theory and Applications*. 2011, San Francisco: Elsevier Science & Technology.
- [6] Aljazzar, H. and S. Leue, *K *: A heuristic search algorithm for finding the k shortest paths*. Artificial intelligence, 2011. **175**(18): p. 2129-2154.
- [7] Garcia Diez, S., J. Laforge, and M. Saerens, *Rminimax: An Optimally Randomized MINIMAX Algorithm*. IEEE Transactions on Cybernetics, 2013. **43**(1): p. 385-393.
- [8] Abbasi, M. and M. Rafiee, *Efficient parallelisation of the packet classification algorithms on multi-core central processing units using multi-threading application program interfaces*. IET computers & digital techniques, 2020. **14**(6): p. 313-321.
- [9] Abdelbar, A.M., Alpha-Beta Pruning and Althöfer's Pathology-Free Negamax Algorithm. Algorithms, 2012. 5(4): p. 521-528.
- [10] Millington, I., Artificial intelligence for games. 2nd ed. ed, ed. J.D. Funge. 2009, Boca Raton, Florida ;: CRC Press.



THE UNIVERSITY
of ADELAIDE