

# Visualization and AI for 3D tic-tac-toe

Student Name: Nan Zhao

Student ID: 201530286

Primary Supervisor: Prof. Boris Konev

Second Supervisor: Prof. Frank Wolter

A DISSERTATION

Submitted to

The University of Liverpool

in partial fulfilment of the requirements  
for the degree of

MASTER OF SCIENCE

# Abstract

Artificial intelligence (AI) is very popular in the computer games, such as Chess, Tic-tac-toe, and Connect-4. The project is a random game about visualization and AI for 3D tic-tac-toe. AI firstly starts or player firstly starts in the game. If a player uses two pieces (X and O), AI uses two pieces (Y and W) to play game with a player, the winner gets 2 scores. If a player uses one piece (X or O), AI uses one piece (Y or W) to play game with a player, the winner gets 2 scores. Only having the same four pieces become a line from row, column, diagonal wins the game, finally, those scores are summed up, player gets higher scores will win the game. The goals of the project have two purposes, the one is that the game of tic-tac-toe uses the min-max with alpha-beta pruning optimized, because alpha-beta is helpful to find a better location of piece on the chessboard. The time of performed iterations is much less than using only mini-max algorithm. The other is that using pygame of the 3D game engine implements 4x4x4 chessboard of 3D cube and creates four 4x4 chessboard of 2D flat.

The experience accomplishes the application of minimax algorithm and the visualization of 2D and 3D tic-tac-toe, using pygame of the 3D game engine creates user interface friendly. The accomplishment of the project is original ideas, AI and player both sides have two pieces respectively to play tic-tac-toe on the four 4x4 chessboard, and the game uses min-max algorithm with alpha-beta pruning optimized, using pygame of 3D computer game engine creates 2D and 3D intuitive user-friendly interface. At the beginning, the project uses Unity to complete visualization by using python programming, however, after searching the amount of materials, it is found that Unity usually uses C programming and not supports for python. Because the project uses python. Therefore, the game uses pygame to achieve visualization and create user interface friendly instead of Unity. It is easily to find that AI always defeats human through playing the game of tic-tac-toe.

# Student Declaration

I confirm that I have read and understood the University's Academic Integrity Policy.

I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the program of study.

I confirm that I have not copied material from another source nor committed plagiarism nor fabricated data when completing the attached piece of work. I confirm that I have not previously presented the work or part thereof for assessment for another University of Liverpool module. I confirm that I have not copied material from another source, nor colluded with any other student in the preparation and production of this work.

I confirm that I have not incorporated into this assignment material that has been submitted by me or any other person in support of a successful application for a degree of this or any other university or degree-awarding body.

SIGNATURE \_\_\_\_\_Nan\_Zhao\_\_\_\_\_

DATE           September 15, 2022

# Acknowledgments

Firstly, I am very grateful for Prof. Boris Konev of primary supervisor, he is very responsible and friendly. At the beginning, I confused how to implement the visualisation, Prof. Boris gave me lots of suggestion and we discussed more than one hour by online. He encouraged me that my design was interesting, especially for both players respectively two pieces. I got lots of confident through communicating with Prof. Boris. I also grateful for the Prof. Frank Wolter of second supervisor, he gave me feedback of final presentation, which can help me to write my dissertation better. I am grateful for the University of Liverpool, University of Liverpool gives me an opportunity to study acknowledge of computer science, I was energy engineer in the past, I am very interested in computer science through learning a year. Although I graduate, it is new beginning for me. Through learning a year, I decide to continue study in AI and plan to apply for ph.d.

Finally, I am grateful my parents, they always support my ideas and encourage me to do things that I am interested. I arrived into the UK in the last year, I missed my parents and my hometown, my parents told me that I need be independent. Gradually, I had some new friends and I found that I like culture of UK more and more. Liverpool is a famous port city in north-west England and is one of European Capital of Culture. The residents of Liverpool are quite ethnically rich, with the oldest black community in the UK and the oldest Chinese community in Europe. The Guinness Book of World Records named Liverpool the "pop capital of the world" [8], where the Beatles, Billy Free, Gerry and the Leaders and many others made their fortunes. Two Premier League clubs: Liverpool and Everton are also based in Liverpool. When having free time, watching football game is a very relaxing thing. The University of Liverpool has abundant resources, the library opens in all day except holidays, and the laboratory has advanced hardware equipment, I am lucky that I can study there.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	2D tic-tac-toe . . . . .	1
1.2	3D tac-tic-toe . . . . .	2
1.3	Mini-max algorithm with alpha-bate pruning . . . . .	3
1.4	Outcome . . . . .	4
<b>2</b>	<b>Aims and Objectives</b>	<b>5</b>
<b>3</b>	<b>Background</b>	<b>6</b>
<b>4</b>	<b>Design</b>	<b>10</b>
4.1	Original design . . . . .	10
4.2	Changes to original design . . . . .	10
<b>5</b>	<b>Implementation</b>	<b>11</b>
<b>6</b>	<b>Evaluation Key Points</b>	<b>13</b>
6.1	Evaluation . . . . .	13
6.2	Professional Issues . . . . .	13
6.3	Learning Points . . . . .	13
<b>7</b>	<b>Conclusion</b>	<b>15</b>
	<b>Bibliography</b>	<b>16</b>
<b>A</b>	<b>Some Interesting Bit of Code</b>	<b>17</b>
<b>B</b>	<b>Experimental results</b>	<b>18</b>
<b>C</b>	<b>Original design document</b>	<b>19</b>

# List of Figures

1.1	A winning example of the player with a cross (X) . . . . .	1
1.2	The draw of tic-tac-toe . . . . .	1
1.3	The first player is in the corner with the first stone . . . . .	2
1.4	A minimax tree example . . . . .	3
1.5	An illustration of alpha-beta pruning . . . . .	4
2.1	4x4x4 cube for the 3D tic-tac-toe . . . . .	5
3.1	Comparison two algorithms . . . . .	6
3.2	an example of pygame application . . . . .	7
3.3	graphical display with the title . . . . .	9
5.1	4x4x4 cube of 3D tic-tac-toe . . . . .	11
5.2	four 4x4 flat of 2D tic-tac-toe . . . . .	12
6.1	Comparison editors of different engines . . . . .	14
A.1	some part interesting coding . . . . .	17
B.1	4x4x4 cube of 3D tic-tac-toe . . . . .	18

# Chapter 1

## Introduction

### 1.1 2D tic-tac-toe

The game of tic-tac-toe began from the ancient Egypt and found around 1300 years on the roof tiles[1]. The early changing of tic-tac-toe started at Roman Empire around the first century BC[2]. The tic-tac-toe structure is simple, so it became general example of artificial intelligence(AI). It was easy to conclude the winning rule of game and reduced a process from strategy, which was played both computer and users.

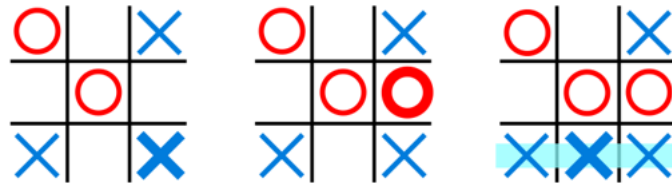


Figure 1.1: A winning example of the player with a cross (X)

There are two players, one of player has a piece of X and the other is a piece of O, they take turns playing their own piece on the 3x3 grid. The first line up of a row, a column, a diagonal and will be a winner of the game(as shown figure 1.1). If two players are correct, the chessboard will be full and the the game will be tied(as shown figure 1.2).

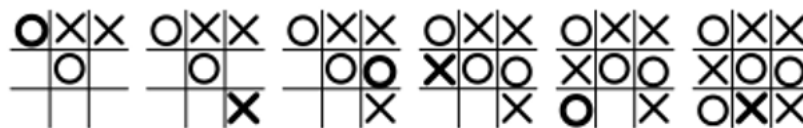


Figure 1.2: The draw of tic-tac-toe

The game actually is controlled by the first player, the first player is the offense and the second player is the defender. If the first player and the first piece are in the corner, the chance of wining is the largest(as shown figure 1.3). If the second player who has one piece is in the side of grid, the first player could use two lines to include the second player, so the second player must put piece into the center of the grid. If the first player puts pieces into the middle of the grid, the second player must put piece into the corner of the grid, so the second player will not lost the game. If the first player puts piece into the side of the grid, the second player could put piece into the center or the corner of the grid, or putting the opposite of the first player.

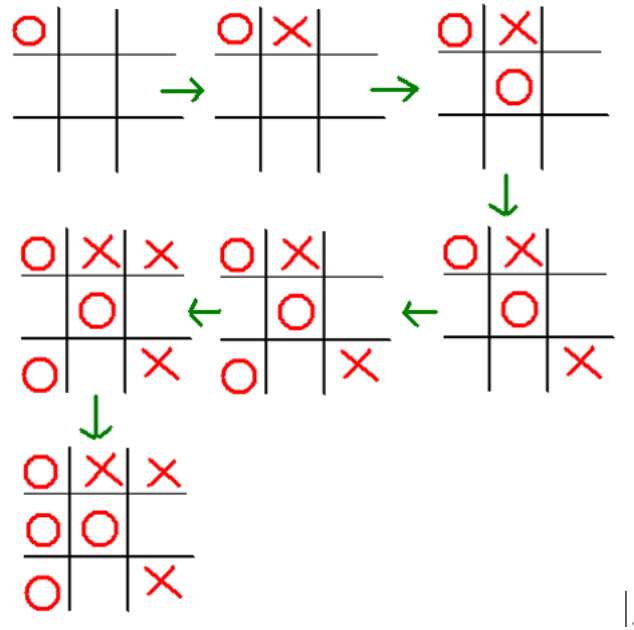


Figure 1.3: The first player is in the corner with the first stone

## 1.2 3D tic-tac-toe

3D tic-tac-toe is also called Qubic, which is an abstract strategy chessboard game. Generally two players play the each other. 3D tic-tac-toe is analogous with traditional tic-tac-toe, but showing on the cube array, usually using 4x4x4. Player in turn puts pieces into blank cells on the array, the first player continuous getting four marked cells will win the game, winning row could be horizontal, vertical or diagonal on a single board(such as common tic-tac-toe), or vertical on the column, or through diagonal of four checkerboard. There are commonly two types of tic-tac-toe one type is 3x3x3 and two players, the other is 4x4x4 and two players. In the 3x3x3 and two players, the game can not draw at the end, and the first player easily wins the game except preventing the first player from putting piece into the center cell, in this situation the second player easily wins the game. The first player easily wins the game through fully banning using the center cell. The perfect game will draw through including the third player. The game will be fair and all of players will win through player using the side randomly, but this is affected by the chance. The game will be fair through randomly using piece(X or O) and all of players will win the game[3]. In the 4x4x4 and two players, there are 76 lines of winning. On the four 4x4 board or each horizontal flat board, there are four columns, four rows, and two diagonals, totally summing up 40 lines. There are 16 vertical lines, each line rises from a cell of bottom board through cells of other boards corresponding to rising. There are 8 vertical flat that are parallel to side of board, each flat increases two diagonals. Finally, there are two vertical flat, including 4x4 diagonal of board, each flat contributes to two diagonals and each diagonal includes two corner and two inside cells. There are 16 cells located behind



four lines, this is 8 corner cells and 8 inside cells, they respectively include 7 different winning lines. Other 48 cells that are 24 front cells and 24 edge cells respectively include four winning lines[4].

### 1.3 Mini-max algorithm with alpha-beta pruning

Mini-max belongs to one of algorithm, it is usually used to make strategy and game theory to find the best move of player, assuming the opponent also being the best move[5]. Mini-max algorithm usually is used to two players turn playing game, such as tic-tac-toe, chess, etc. In the mini-max algorithm, one of players is called maximizer, the other is called minimizer, maximizer tries to get highest score comparing with the minimizer getting lowest score. Conversely, if the minimizer gets highest score, the maximizer would get lowest score. For each giving state, if the maximizer have the advantage over the minimizer, the chessboard would get some positive score, if the minimizer have the advantage over the maximizer, the chessboard would get some negative score[8].

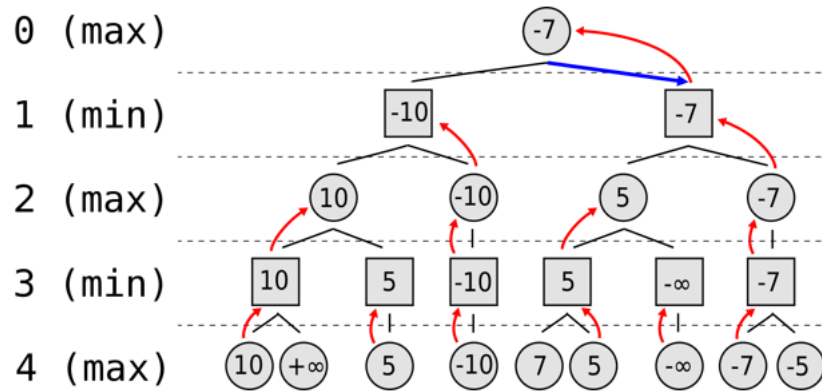


Figure 1.4: A minimax tree example

It is better understanding the minimax algorithm from an example(as shown the figure 1.4), each player of playing the game has at most two possible move in every game. Algorithm products tree on the right side, circle means that the move of running algorithm player(maximize player), square means opponent's the move(minimize player). The tree is limited to four moves because computing resource limits. The minimax algorithm uses heuristic evaluation function to evaluate each leaf node and obtains obvious value. The maximize player that gets moves is assigned as positive infinity, and the minimize player that gets moves is assigned as negative infinity. In the third level algorithm will choice minimizing child node value for each node, and distribute it to the same node, for example, left side node would choose minimize between 10 and positive infinity, so the value of 10 will assign itself. In the second step, algorithm would choose maximize child node value, those value will be assigned each parents node again, the algorithm continues valuate turn child node both maximize and minimize until it arrives root node, on there it chooses the maximizing move(as shown blue arrow in the figure 1.4). This is move that player should take measure, so

that the loss could decrease as soon as possible[7]. Heuristic value evaluates score of node for favor-ability of maximize player. Therefore, it is higher score for the maximize player to product benefit node comparing with minimize player producing beneficial node. Heuristic value of terminal leaf node(game over) is score of the win,the loss or the draw, this is used to apply the maximize player. Evaluation function evaluates heuristic value of node for non-terminal leaf node of the maximizing searching depth. This quality and search depth of evaluation decide the quality and search depth minimax result[7].

Alpha-beta pruning is a search algorithm, which is used to decrease node of minimax algorithm from search tree. It is confrontational search algorithm and usual used to play two player game for robots(such as tic-tac-toe, chess, connect4, etc). When finding at least one possibility approving the current move is worse than previous move, it stops the move of evaluation. This move does not need evaluate further more. When applying standard minimax tree, it returns to the same move as minimax tree, but it would prune the branch that does not influence on final strategy[8]. In the Fig. 1.5 alpha-beta pruning need not explore grey tree(move from left to right), because known child tree regards an entirety, this would product value of equivalent child tree or worsen value. The maximize and minimize level respectively represent round both player and opponent[9].

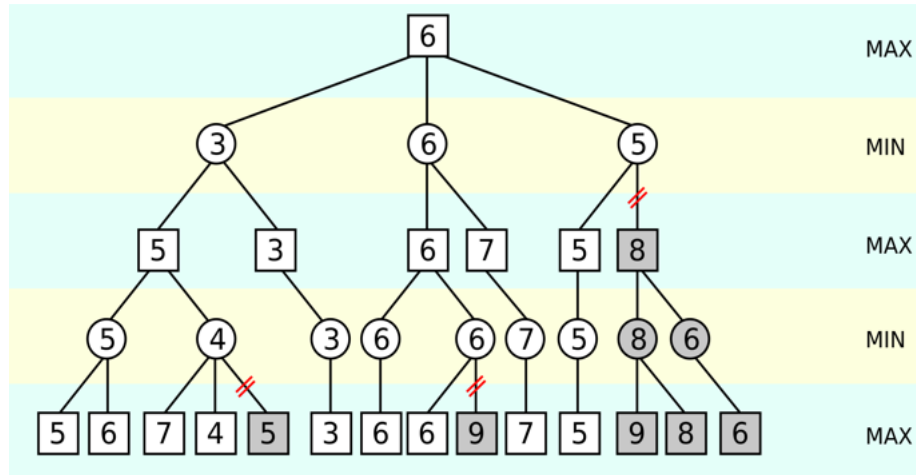


Figure 1.5: An illustration of alpha-beta pruning

## 1.4 Outcome

The project visualizes 4x4x4 cube and creates four 4x4 chessboard of user interface for the game of tic-tac-toe. In the game, AI firstly starts or player firstly starts, this is random. If a player uses two pieces (X and O), AI uses two pieces (Y and W) to play game with a player. If a player uses one piece (X or O), AI uses one piece (Y or W) to play game with a player. Only having the same four pieces become a line from row, column, diagonal will get one score, finally, summing up those scores, getting higher scores will win the game. Actually, AI easily wins the game.

## Chapter 2

# Aims and Objectives

In the project, AI and player both sides have two pieces respectively to play tic-tac-toe on the four 4x4 board. Player has two pieces of X and O, player B has two pieces of Y and W. The aims and objectives have two purposes, the one uses min-max algorithm with alpha-beta pruning optimized, the other is that visualization the cube uses pygame of 3D computer game engine and creates an intuitive user-friendly interface(as shown figure 2.1).

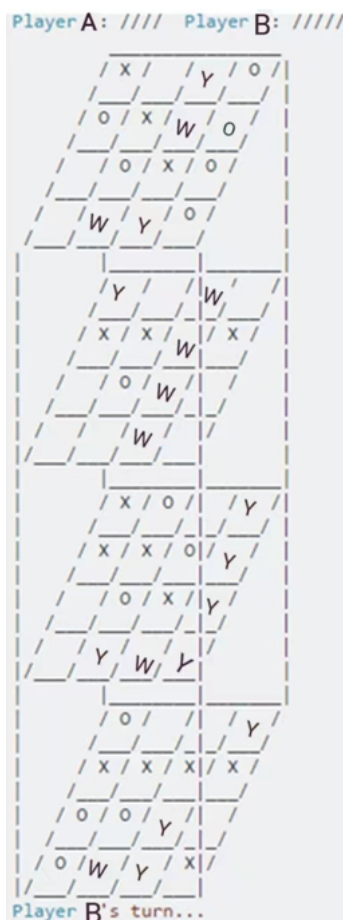


Figure 2.1: 4x4x4 cube for the 3D tic-tac-toe

## Chapter 3

# Background

The project mainly uses minimax algorithm to visualise 4x4x4 cube achieve and create user interface of four 4X4 flat board by playing the game of tic-tac-toe. Therefore, I read some reviews about minimax algorithm[5] and visualization from other authors[10].

Algorithm Used	Mini Max		Alpha-Beta Pruning	
Difficulty Level	No. of Iterations performed	Computation Time (ms)	No. of Iterations performed	Computation Time (ms)
Depth 1 (Easiest)	7	0.00	7	0.00
Depth 4 (Normal)	2799	33.00	477	6.00
Depth 8 (Hardest)	5847005	55441.00	71773	1009.00

Figure 3.1: Comparison two algorithms

There is comparing minimax algorithm and minimax algorithm with alpha-beta pruning on the figure 3.1, it observed that user plays game in the single mode and uses minimax algorithm, and then AI need 7 times iteration the computation time is 0 ms, depth is 1. But using minimax algorithm with alpha-beta pruning in the same difficulty, it also uses 7 times iteration and the computation time is 0 ms. With the difficulty level increasing, AI need 2799 times iteration by using minimax algorithm on the depth 4 of difficulty level, the computation time is 33 ms, but using minimax algorithm with alpha-beta pruning in the same difficulty, it also uses 477 times iteration and the computation time is 6 ms. On the depth 8 of difficulty level, AI need 5847005 times iteration by using minimax algorithm, the computation time is 55441 ms, but using minimax algorithm with alpha-beta pruning in the same difficulty, it also uses 71773 times iteration and the computation time is 1009 ms[5].

Programming language has lots of framework and libraries to create anything from simple game to large complex game. For example, Python is also used to development of game. There are lots of libraries or frameworks for development of game in python. For example, Arcade is a library that create 2D style of video game in python. Pyglet is the window of python and multimedia library, and also is used to development game. Cocos2d is framework

of 2D game building on the pyglet. PyODE is open source of Open Dynamics Engine in python, this is an open physical engine. Pymunk is a physical library of using 2D physical library, and uses python to finish 2d rigid body physics, it usually is used to game, demonstrate, or physic of application program. Pybox2d is 2D physical library that is used to game and simple simulation, it is based on the Box2D library of C++ programming and supports lots of shapes. Pygame is one of the most common library in Python program for create different games. Pygame is mainly described because the game of tic-tac-toe use pygame to implement.

```
import pygame

def main():
    print('Starting Game')

    print('Initialising pygame')
    pygame.init()  # Required by every pygame application

    print('Initialising HelloWorldGame')
    pygame.display.set_mode((200, 100))
    pygame.display.set_caption('Hello World')

    print('Update display')
    pygame.display.update()

    print('Starting main Game Playing Loop')
    running = True
    while running:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                print('Received Quit Event:', event)
                running = False

    print('Game Over')
    pygame.quit()

if __name__ == '__main__':
    main()
```

Figure 3.2: an example of pygame application

Pygame is a python library of cross-platform, free, open source, this can easily build game and multimedia applications. The development of python started on October 2000 year, pygame built over the SDL(simple direction media layer) library, SDL is a cross-platform development library, this can provide audio, keyboard, mouse, joystick and graphics hardware with access by OpenGL and Direct3D. Pygame also supports many kinds of additional backend including WinDIB, X11, Linux Frame Buffer etc, this can promote portability. SDL officially supports Windows, Mac OS X, Linux, iOS and Android. SDL uses C program to write and pygame provides a wrapper around SDL, but adding function of pygame is not found in the SDL, it can easily create graphic or video game. Those functions include vector math, collision detection, 2D sprite scene graph management, MIDI support, cameras, pixel array manipulation, advanced free font support.

There is an example of hello world about Pygame application. There are some key steps when using pygame(as shown figure 3.2).

Firstly, import pygame,because pygame is not one of default modules in python, so firstly import pygame. The statement of import pygame would import pygame into the code,and use function and class from pygame.

Secondly, initialise pygame, almost each pygame module need initialise, the simplest method transfer pygame.init(), this can install pygame environment in order to provide what need. If missing call the function, the function will get the error message, such as pygame.error:video system not initialised (or other similar error). If getting the method, checking whether transfer pygame.init() or not. It usually initialise single pygame module(such as pygame.font module), but pygame.init() is the most common method for setting up pygame.

Thirdly, setting up the display, after initialise pygame framework, setting up the display. The display uses pygame.display.setmode() function in the code, function receives a tuple that creates size of window(the height of pixel is 100 and the width is 200 in the example). It notes that trying to afferent two parameters in order to transfer function instead of tuple, this shows error,this function returns drawing surface or screen/window games for showing project, such as graphic,message, shape etc. The example is simple in the figure 3.2, so it need not save in the variable, but other examples need do it instead of the example.Setting up window/framework title(or title), it shows window on the title.

Fourthly, render the display. It need transfer pygame.display.update function, this function plots current display details. It is a blank window at the moment, but execution in the background updates a series of the display in the game,when program executes, updating display and transfer function. The batch process a series of updates in order to update display. It can instruct the part of display that need redraw instead of redrawing all of the window, this instructs redrawing rectangle by parameters transferring update() function. However,the example is simple, so it can redraw the whole of window and need not parameters transferring function.

Fifth, Main loop of game. It is common to have a main game loop, and it can drive users to deal with inputting, modify state of game, and update display. In the example, while running is loop representation, the local variable is initialised as True, it means that while loop ensures game to proceed, until user chooses to quit game by clicking proceed variable to set up False and exiting the loop. The loop can transfer update() to update display in many cases. The example does not do it because the display does not change anything.

Sixth, Monitor event for driving the game. As shown above, event queue allows users to input queue, the the game deals with it. In the figure 3.1, receiving the for loop can uses pygame.event.get(), and then check whether event is a pygame. QUIT event.If it is, it sets up execution flag as False. The main While loop game that case will terminate.

Finally, Quit pygame when finished. The module of init() function and quit() function can execute cleaning operation. The program transfers init() at the beginning on the pygame module. So program transfers pygame.quit

at the ending in order to make everything tidied up.

The game of tic-tac-toe uses python programming language, pygame is the module in python. So using pygame accomplishes basic applicable procedure and creates a window with the title 'Minimax algorithm with alpha-beta pruning'(as shown figure 3.3), and then quit the game. Using the exit button on the window quits the program.

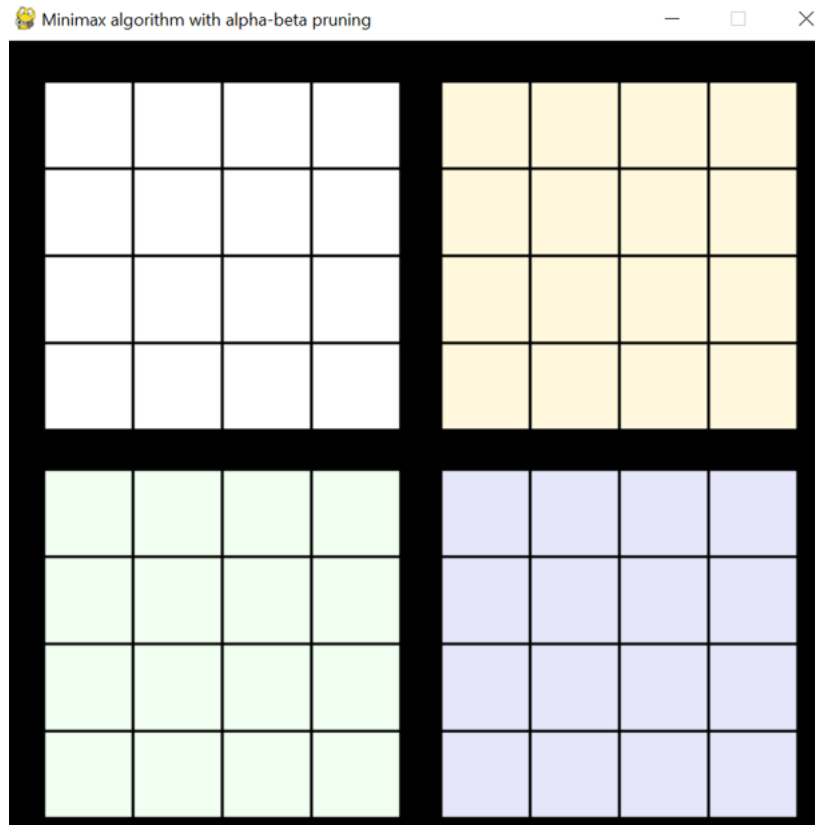


Figure 3.3: graphical display with the title

# Chapter 4

## Design

### 4.1 Original design

2D tic-tac-toe and 3D tic-tac-toe have described in the part of introduction. AI easily defeat human in 2D and 3D tic-tac-toe. In the project, Using 4x4x4 dimension extends chessboard in order to increase interest of the game, there are total 76 lines of winning the game. Although the AI must be wining, the difficulty would increase. At the same time, AI and human both side respectively have two pieces. When having four pieces becomes a line, the game would win. For example, player A has two pieces of X and O, player B have another two pieces of W and Y, the order of putting pieces is random, this means that player A or player B firstly starts at the beginning. If player A uses one piece(X or O), player B also uses one piece(Y or w), if player A uses two pieces of X and O, player B also uses two pieces of W and Y, this can keep fair the game. If a line is OXOO four pieces, player A can not win the game, because the four pieces is not the same shape, Only four pieces of XXXX or OOOO become a line, player A can win the game. For the player B, only four pieces of YYYY or WWWW become a line, player B can win the game. Each player win one time of the game and then get one score, the score will be accumulated, finally, play of getting higher score will win the game. The project need uses minimax algorithm and accomplish visualization, using the Unity implements the visualization, because Unity is has lots of advantages. For example, Unity has visualization environment and cross-platform support, the visualization environment not only includes visualizing modeling tool, but also has integrated environment and building chain, this is helpful for developer to improve productive forces. Unity has modeling system that is used to build game object, this is different from inherit, the object of Unity is built by combined function blocks instead of putting the node of inherit tree, this way is easy to build the prototype of game[11].

### 4.2 Changes to original design

In the game, using python program to finish, so choose to use pygame library. Using mouse randomly operates pieces on the chessboard. There are some key steps when using pygame. The project uses python programming language instead of C programming language, because Unity uses with C instead of Python. I use pygame to visualize.



## Chapter 5

# Implementation

In the project, I use minimax algorithm with alpha-beta to implement 4x4x4 cube of 3D tic-tac-toe and visualise the cube using pygame(as shown figure 5.1). I also create an intuitive user-friendly interface, it consists of four 4x4 flat square(as shown figure 5.2).

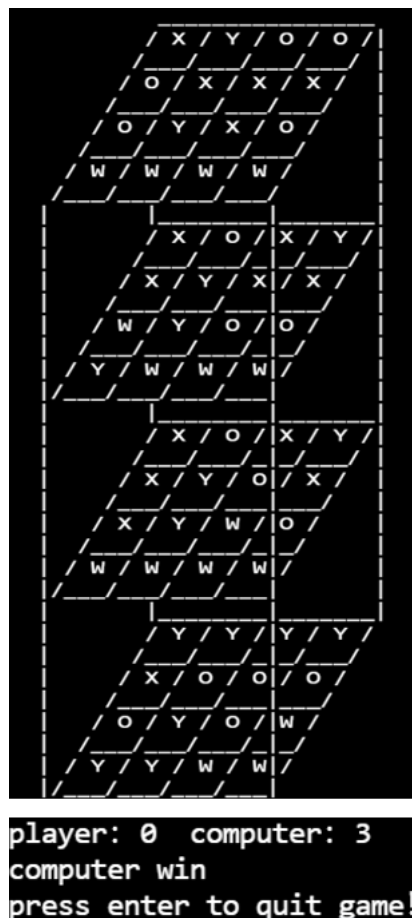


Figure 5.1: 4x4x4 cube of 3D tic-tac-toe

This is a random game of tic-tac-toe, AI firstly starts or player firstly starts in the game, If a player uses two pieces (X and O), AI uses two pieces (Y and W) to play game with a player. If a player uses one piece(X or O), AI uses one piece(Y or W) to play game with a player. Only having the same four pieces become a line from row, column, diagonal will get one score, finally, summing up those scores, getting higher scores will win game. In the game, I use the mini-max with alpha-beta pruning optimized, because alpha-beta is helpful to find a better location of piece on the board.

The time of performed iterations is much less than using only mini-max algorithm. Min-max algorithm is like zero sum game, AI and player play against in the tic-tac-toe game where mini-max algorithm is used to predict the state of game. Player tries to predict the opponent's next move and try to minimize the function, and player always tries to maximize own functions based on the current situation in order to win the game. Heuristic function plays an important role in strategies of Minimax. Each node of the tree will have a heuristic function associated with it. Based on this heuristic function, it will decide to move towards the nodes that are most beneficial to them.

Top-Left Board (White)				Top-Right Board (Yellow)			
X	Y	O	O	X	O	X	Y
O	X	X	X	X	Y	X	X
O	Y	X	O	W	Y	O	O
W	W	W	W	Y	W	W	W

Bottom-Left Board (Light Green)				Bottom-Right Board (Light Purple)			
X		X	Y	Y	Y	Y	Y
X	Y	O	X	X	O	O	O
X	Y	W	O	O	Y	O	W
W	W	W	W	Y	Y	W	W

Figure 5.2: four 4x4 flat of 2D tic-tac-toe

When both sides put pieces on the full of chessboard(as shown figure5.2), player and AI will get scores of the game (as shown figure 5.1), computer gets 8 scores, player get 0 score. The rule of getting score include two aspect, the one is that player uses one pieces(X or O), AI uses one pieces(Y or W), the winner will get 1 score, the other is that player uses two pieces(X and O), AI uses one pieces(Y and W), the winner will get 2 scores. Finally, summing up those scores, getting higher scores will win the game. It shows from figure 4.1 that player gets 0 score,because there is no the same four pieces connecting a line,computer gets 3 scores,because there is double four WWWW pieces in a lines and a four YYYYY pieces in a line, so computer win the game. If wanting to quit the game,pressing the enter key.

## Chapter 6

# Evaluation Key Points

### 6.1 Evaluation

The achievement of project is that I use python program and pygame library to implement the project, and I also visualise 4x4x4 cube, I create user interface friendly on the four 4x4 board by using minimax algorithm with alpha-beta pruning optimized. The shortfall of project is that I do not use Unity to create user interface due to non-compatible python.

The change of the project is the way of visualization, at the beginning, I want to use Unity to complete visualization by using python, I found that Unity usually uses C programming and not support python, I want to use python in the project. Therefore, I choose to use pygame to achieve visualization and create user interface friendly instead of Unity.

### 6.2 Professional Issues

My coding obeys to the Code of Conduct of British Computer Society (BCS), I write my own coding according to my ideas, this is very objective.

### 6.3 Learning Points

I obtain the knowledge of mini-max algorithm, alpha-beta pruning, game engine through implementing the project. Minimax is a decision rule in AI, decision theory, game theory, statistics and philosophy, it is used to minimize possible loss in the worst-case (maximum loss) scenarios. When dealing with gaining, it is also called maximize minimum gain. Minimax initially is made for the multiplayer zero-sum game theory, it includes scenarios where players move alternately, and simultaneously and extends more complex game theory and exists uncertain general decision. Alpha-beta pruning is a search algorithm, it applies number of nodes of search tree for the Minimax. Alpha-beta pruning is an adversarial search algorithm and applies to two players (such as tic-tac-toe, chess, Go). When the Minimax algorithm assesses the subsequent moves of the strategy are worse than those of the previous strategy, subsequent development of the strategy was halted. Alpha-beta pruning is similar with Minimax but prunes the branches of non-influence final decision. There are different game engines. Pygame is a set of cross-platform Python modules designed for video games, which contains images and sounds. Pygame is built on top of SDL, it allows real-time video game development without being bound by low-end languages such as C or even lower-end assembly languages. Based on such an assumption, all required game functions and concepts completely simplify the game logic itself, and

all resource structures can be provided by high-level languages such as Python. Unity is the game engine by development of Unity Technologies, it is used to create 3D and 2D game, the unity engine was shown on the Apple Worldwide Developers Conference in the 2005 year. Unity engine has system of modules based on components in the game. Advantages of Modular Approaches can increase flexible, unity allows efficient and trouble-free component to match. the framework of Unity is used with the C program, Unreal engine uses with C++ program, CryEngine uses the C or C++ program. There is comparing about the three engines. Unity provides minimum number of editor because some function have entrusted main editor tab. Virtual engine is only one having supporting actual editor engine that includes AI elements and supporting design of user interface. CryEngine framework uses tool of 3D object modeling and environment editor of environment management, lots of engine function including whole of editor can use form of library or plugin to add the engine(as shown figure 6.1).

Unity engine has better document in the comparing engine, most of engines describe deeply and the document also include lots of examples of engine function, this is very helpful for new users. Virtual engine document writes well the same unity, but still find some drawback. Document of CryEngine engine is most worsen than Unity and Unreal engine. The drawback of document is that the part of document use engine of the old version to write since it is not updated. Unity is an engine having lots of function, it can create own 2D and 3D game for many genres. Unity is beneficial to the special mode of Unity2D, this is used to create 2D game and the best engine in the 2D engines. Virtual engine can create 2D game, but this engine is not too intuitive and it is more complex. Virtual engine and Unity engine are used to create almost all types of games, including moving platform, the function is very wide. CryEngine is not intuitive for create 2D game and can not finish implementation of higher level mechanisms. CryEngine can exert own all potential game in the implementation of FPS, and can create own 3D engine, this engine can achieve similar with FPS game genre that is third person game and can not product difficulty for user[11].

Editors	Unity 2018	Unreal Engine 4	CryEngine 5.5
Basic	Unity Editor	Level Editor	Sandbox Editor
Materials	X	Material Editor	Material Editor
IDE/Scripts	Visual Studio, MonoDevelop	BluePrint Editor, Visual Studio	FlowGraph, Visual Studio
AI	X	BehaviorTree Editor	X

Figure 6.1: Comparison editors of different engines

Unity engine can provide guidance system for only one supporting user, which can achieve NPC behavior. The guidance system allows non-player to avoid moving and marking location for the navigation shortcuts. Virtual engine can better support mechanism of NPC behavior. Unreal is similar with Unity, and provides behavior tree tool and environment query system. EQS is dynamic system that allows users to execute, and this is suitable to object query of searching condition. CryEngine is the same as other engines, and has own independent character navigation system[11].

## Chapter 7

# Conclusion

In the project, the game of tic-tac-toe uses minimax algorithm with alpha-beta pruning optimized to implement 3D cube of 4x4x4 and 2D four 4x4 chessboard. Using pygame library in python visualises the user interface friendly. Both sides respectively have two pieces instead of one pieces, AI has two pieces of Y and W, human has two pieces of X and O. If human uses one piece(X or O), AI also uses one piece(Y or W), the winner will get one score. If human uses two pieces(X and O), AI also uses two pieces(Y and W), the winner will get two scores, this ensures the fair of the game. Finally, summing up the score, getting higher scores will win the game. The interest of tic-tac-toe increases by using two pieces respectively for both players.

# Bibliography

- [1] Zaslavsky, Claudia. Tic Tac Toe: And Other Three-In-A Row Games from Ancient Egypt to the Modern Computer. *Crowell*. 1982.
- [2] Parker,Marla. She Does Math!: Real-life Problems from Women on the Job *J. Mathem -atical Association of America*. pp. 153[2020-06-08],1995.
- [3] Golomb,Solomon W. Hypercube Tic-Tac-Toe. More Games of No Chance. *More Games of No Chance* **42**, 2002.
- [4] R. Silver.The group of automorphisms of the game of 3-dimensional ticktacktoe *Amer. Math. Monthly. Mathematical Association of America*. **74**, pp. 247–254, 1967.
- [5] R. Nasa1, Alpha-Beta Pruning in Mini-Max Algorithm–An Optimized Approach for a Connect-4 Game *International Research Journal of Engineering and Technology (IRJET)* **05**, pp.1637-1641, 2018.
- [6] Russell, Stuart J.Artificial Intelligence: A Modern Approach (2nd ed.) *Upper Saddle River, New Jersey: Prentice Hall*, pp.163–171, 2003.
- [7] Osborne, Martin J.A Course in Game Theory (print ed.) *A Course in Game Theory (print ed.)*, 1994.
- [8] Russell, Stuart J.Artificial Intelligence: A Modern Approach (3rd ed.). *Upper Saddle River, New Jersey: Pearson Education, Inc* **167**,2010.
- [9] Pearl,Judea. The Solution for the Branching Factor of the Alpha-Beta Pruning Algorithm and Its Optimality *Communications of the ACM* **25**, pp. 559–64,1982.
- [10] Hunt,J.Introduction to Games Programming. In: Advanced Guide to Python 3 Programming. Undergraduate Topics in Computer Science. *Springer,Cham* **12**, pp. 121–150, 2019.
- [11] Barczak A M, Woźniak H. Comparative study on game engines. *tudia Informatica. Systems and Information Technology. Systemy i Technologie Informacyjne* **12**, pp. 1–2, 2019.

# Appendix A

## Some Interesting Bit of Code

I might include some particularly interesting part of my code here that is referred to elsewhere in the document.

```
class SwapGridData(object):

    def __init__(self, grid_data):
        self.grid_data = list(grid_data)
        self.grid_size = calculate_grid_size(self.grid_data)

    def x(self):
        return join_list(x[::-1] for x in split_list(self.grid_data, self.grid_size))

    def y(self):
        group_split = split_list(self.grid_data, pow(self.grid_size, 2))
        return join_list(join_list(split_list(x, self.grid_size)[::-1] for x in group_split))

    def z(self):
        return join_list(split_list(self.grid_data, pow(self.grid_size, 2))[::-1])

    def reverse(self):
        return self.grid_data[::-1]

def calculate_grid_size(grid_data):
    return int(round(pow(len(grid_data), 1.0 / 3.0), 0))
```

Figure A.1: some part interesting coding

## Appendix B

# Experimental results

Here I give details of some experiments that were performed during the course of the project.

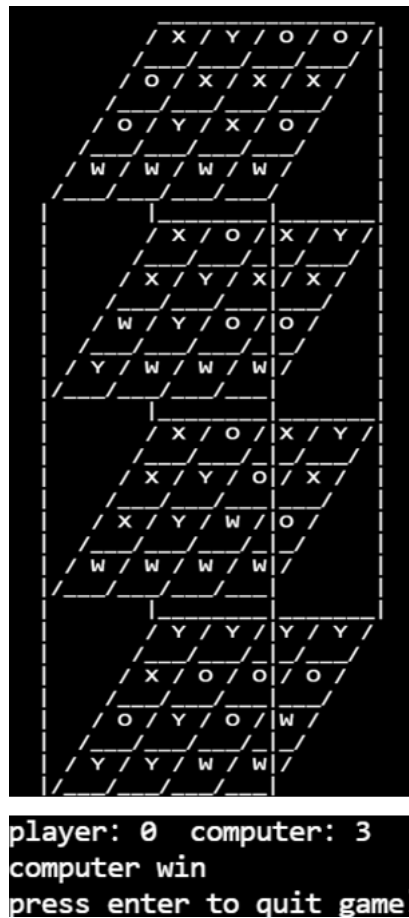


Figure B.1: 4x4x4 cube of 3D tic-tac-toe



## Appendix C

# Original design document

Here I might insert the original design document for the project, in order to refer to it and changes that I made.

2D tic-tac-toe and 3D tic-tac-toe have described in the part of introduction. AI easily defeat human in 2D and 3D tic-tac-toe. In the project, Using 4x4x4 dimension extends chessboard in order to increase interest of the game, there are total 76 lines of winning the game. Although the AI must be wining, the difficulty would increase. At the same time, AI and human both side respectively have two pieces. When having four pieces becomes a line, the game would win. For example, player A has two pieces of X and O, player B have another two pieces of W and Y, the order of putting pieces is random, this means that player A or player B firstly starts at the beginning. If player A uses one piece(X or O), player B also uses one piece(Y or w), if player A uses two pieces of X and O, player B also uses two pieces of W and Y, this can keep fair the game. If a line is OXOO four pieces, player A can not win the game, because the four pieces is not the same shape, Only four pieces of XXXX or OOOO become a line, player A can win the game. For the player B, only four pieces of YYYY or WWWW become a line, player B can win the game. Each player win one time of the game and then get one score, the score will be accumulated, finally, play of getting higher score will win the game. The project need uses minimax algorithm and accomplish visualization, using the Unity implements the visualization, because Unity is has lots of advantages. For example, Unity has visualization environment and cross-platform support, the visualization environment not only includes visualizing modeling tool, but also has integrated environment and building chain, this is helpful for developer to improve productive forces. Unity has modeling system that is used to build game object, this is different from inherit, the object of Unity is built by combined function blocks instead of putting the node of inherit tree, this way is easy to build the prototype of game[11].